JPC

PPCPC

OCTOBRE

1983

VOLUME-1

NUMERO-7

LE NUMERO 20FF

directeur de la publication: Philippe GUEZ

APPLICATIONS

JEAN FRANCOIS PELANNE	15	2 NOUVELLES FONCTIONS "H--" & "2--"
ROBERT SCHWARTZ	19	RECENSEMENT DES BUGS
ROBERT SCHWARTZ	19	BIZARRE, VOUS AVEZ DIT BIZARRE ?

LES PREMIERS PAS DE BEBE

PHILIPPE GUEZ	1	LES REGISTRES ⊢ & e
PHILIPPE GUEZ	3	LES REGISTRES a & b
PHILIPPE GUEZ	4	LE REGISTRE c
PHILIPPE GUEZ	5	LE REGISTRE d
ROBERT SCHWARTZ	12	REG M, N, O, P & ALPHA
ROBERT SCHWARTZ	19	Robert tu patines dans la choucroute

PROGRAMMES

ROBERT SCHWARTZ	6	41C/CV	DH
JEAN BOSCHAT	13	41C/CV	SFA-RFA (SAVE-RCL ASCII)
SCHWARTZ - PPC ROM	20	41C/CV	SK et RK (Suspendre et régénérer les ASN)
SCHWARTZ - SP Made easy	26	41C/CV	LBX (LB avec XFCTION)
SCHWARTZ - PPC ROM	5	41C/CV	IF (INVERT FLAG)
SCHWARTZ - SP Made Easy	28	41C/CV	KAX - MKX
JACQUES VAUCELLE	34	41C/CV	JVP
SCHWARTZ - PPC ROM	36	41C/CV	PD - QR - 2D
ROBERT SCHWARTZ	36	41C/CV	PP
DAMIEN DEBRIL	18	75C	TELEPHONE

PETIT THEATRE DES MICROCODES

PHILIPPE GUEZ	22	CODES RETOURNES PAR C = KEY
PHILIPPE GUEZ	23	NOM D'UNE FONCTION, PROMPT, ...
PHILIPPE GUEZ	23	CODES CARACTERE
PHILIPPE GUEZ	24	NON PROGRAMMABILITE, NPIE

LA REVUE DES CODES BARRES

ROBERT SCHWARTZ	14	SFA - RFA
ROBERT SCHWARTZ	28	LBL DELOCALISES EN CB ET FCT SUR DEUX OCTETS
ROBERT SCHWARTZ	29	CB POUR DONNEES ALPHA ET APPEND ALPHA
GILBERT TISSERAND	39	FONCTIONS DES MODULES HORLOGE, PLOTTER, LECTEUR
GABRIEL GIL	42	XROM DE 4,Ø à 7,63

DIVERS

CHOCS EN RETOUR:	ROBERT SCHWARTZ	6 & 7 JPC V1N5
	ROBERT SCHWARTZ	19 HN
WOLFGANG BALTES	9	CHERS AMIS (CCD ROM)
ROBERT SCHWARTZ	15	LETTRE OUVERTE A DANIEL JACOB
ROBERT SCHWARTZ	35	A PROPOS DE COMP
PHILIPPE GUEZ	25	GM
PILOU SANABRIA	25	QUEL MECHANT CE PHILIPPE ?
PILOU SANABRIA	30	HUM ! MIAM ! MIAM !

IMPRIME ET EDITE PAR LE PPC PARIS CHAPTER 56, RUE JEAN JACQUES ROUSSEAU 75001 PARIS - FRANCE

Le PPC PARIS CHAPTER est une association régie par la loi de 1901 , enregistrée à la préfecture de PARIS le 2 décembre 1982 sous le numero 82/3240.

LES PREMIERS PAS DE BEBE

Dans le numéro 4 nous avons vu les registres T, Z, Y, X, L, M, N, O, P, Q; puis 100, 101, A, B, C, D, E, F, G, H, I et J. Soit les 10 premiers registres d'état puis les 12 premiers registres memoire d'une adresse relative supérieure a 99

Voyons maintenant les 5 derniers registres d'état dont l'adresse absolue , nous le rappelons, ne varie pas selon les conditions.

* LES REGISTRES ▶ (APPEND) ET e

Ce sont deux registres utilisé par HP quand une touche est assignée

```
   ASN "AK" 11
 ASN "CODE" 13
ASN "DECODE" 14
 ASN "RCLA" 34
  ASN "STA" 33
          AK
          AK
          AK
          AK
     SIZE 010
       PRKEYS
```

ASSIGNATION DE RCL M ET STO IND N AU TOUCHES 7 ET 8.

CODES HEX:

RCL	=90
STO	=91
M	=75
IND N	=F6
TOUCHE 7	=15
TOUCHE 8	=25

LE CODE A CHARGER A L'ADRESSE ABSOLUE 192 (DEC)

```
F090751591F625
                CODE
197.0000         STA
```

```
ASN "SIZE" 12 "pousse"les REG d'ASN
   ASN "-" 52 } "ouvre" un nouveau
   ASN "+" 53 }  REG d'ASN
192.0000  RCLA
         DECODE
F0044025044115        {"-" ASN à "7"
F090751591F625        {"+" ASN à "8"
              CODE
192.0000       STA
            PRKEYS
```

```
USER KEYS:
 11 AK
 12 SIZE
 13 CODE
 14 DECODE
 21 XROM 01,58
 22 XROM 05,58
 23 XROM 01,63
 24 XROM 05,63
 33 STA
 34 RCLA
 52 XROM 01,53 (RCL M)
 53 XROM 07,54 (STO IND N)
```

C.Q.F.D

```
USER KEYS:
 11 AK
 13 CODE
 14 DECODE
 21 XROM 01,58 (RCL ▶ )
 22 XROM 05,58 (STO ▶ )
 23 XROM 01,63 (RCL e )
 24 XROM 05,63 (STO e )
 33 STA  } RCL et STO la
 34 RCLA } valeur du REG donné ADR ABS en X
              CLX
      0.0000   ***
             RCL '  ( RCL ▶ )
   0.7070-00   ***
            DECODE
00707020390740
             RCL e
     00.0000   ***
            DECODE
00000000007001
    192.0000  RCLA
            DECODE
F0000000917F32
    193.0000  RCLA
            DECODE
F0907F22917A12
    194.0000  RCLA
            DECODE
F0A44B01A45B23
    195.0000  RCLA
            DECODE
F0A44733A44931
    196.0000  RCLA
            DECODE
F0907A02A44821
    197.0000  RCLA
            DECODE
000000000000000
```

Quand vous assignez une touche, certains codes entrent dans les registres d'assignation (dec 192 et suivant) et on charge les 5 digits de droite de ▶ pour les assignations non SHIFTées et les 5 digits de droite de e pour les assignations SHIFTées .

Un registre d'assignation s'écrit toujours de la même façon:

FØ	P1P2 S1S2 T1T2 P3P4 S3S4 T3T4
FØ	toujours présent au début du code a introduire
P	préfix de la fonction (STO, RCL, ISG, ...)
S	suffixe de la fonction (M, d, ..., IND e, ...)
12	première assignation
34	deuxième assignation

Les 5 digits de droite de ▶ servent de mémoire detravail. Il contient le code HEXadecimal de la dernière instruction appelée par l'utilisateur (SST charge 08000, RUN charge 05000, RCL charge 90000, ...).

Comme nous avons vu précédemment, laprincipale fonction de ⊢ est de garder trace des touches assignées non SHIFTées. Quand on presse une touche en mode USER, la machine interroge le drapeau correspondant à cette touche pour voir si elle est assignée. Le cas échéant elle cherche parmi les LBLs ALPHAnumériques et les registes d'assignation pour savoir quel programme ou quelle fonction est assigné cette touche. 36 bits de ⊢ sont réservées à la mise en mémoire; (2 touches font ENTER; la deuxième est tout à fait imaginaire et cette pseudo touche est dotée d'un drapeau. Dans les registres d'assignation, chaque touches possèdent un drapeau (voir tableau cidessous)et même les touches SHIFT et SHIFT SHIFT ont un drapeau malgré quelles ne soient pas réassignables.

La formule donnant le numéro du drapeau en fonction de la touche est:

$$D = 44 - L - 8C$$

L = n° de Ligne de la touche
C = n° de colonne de la touche
D = n° du drapeau

Les digits restant servent de mémoire de travail àla machine. Particulièrement, les digits 4 à1 contiennent le code dela dernière fonction éxécutée.

Aprés l'exécution de PASN (voir le module XFONCTION), l'octet Ø de ce registre contient le code de la touche assignée tel qu'il apparait dans un registre d'ASN

Les 5 derniers digits de e servent de mémoire de travail, les trois derniers contiennent notamment le numéro de ligne de programme. Quand ces trois digits sont à ØØØ, et qu'on est en mode PRGM, on voit à l'affichage ØØ REG abc, (abc est le nombre de registres restant). Quand on arrête un programme au milieu, ou qu'on s'y place par un GTO.xyz, la 41 ignore le n° de ligne et les trois digits contiennent FFF pour que la machine "pense" à calculer le n° de ligne en cas de pression sur R/S ou SST

		COLONNE 1	2	3	4	5
	1	35	27	19	11	3
L	2	34	26	18	10	2
I	3	33	25	17	9	1
G	4	32	(24)	16	8	0
N	5	31	23		15	7
E	6	30	22		14	6
	7	29	21		13	5
	8	28	20		12	4

Relation entre les touches et les numéros de drapeau

* LES REGISTRES a ET b

Ces deux regitres sont utilisés par le pointeur d'adresse et de retour des sous programmes.

Le pointeur adresse utilise un mot de 4 digits (byte) qui sont les 4 derniers du registre b. Si le pointeur est positionné au byte n du registre abc, alors b contient le nombre "ØØ ØØ ØØ ØØ ØØ nabc". Si une sous routine est appelée, l'adresse de retour est stockée dans les 4 digits suivant, à gauche de l'adresse du pointeur, qui est cette fois ci la nouvelle adresse. Achaque appel de sous routine, les adresses de retour sont poussées vers la gauche. Quand le registre b est plein, les adresses sont chargées dans le registre a.

Ils sont de la forme:

|yzab uvwx qrst mn|op ijkl efgh abcd|

registre a registre b

ou efgh est la première adresse de retour; ijkl la seconde; ..., quand un RTN ou un END sont éxécutés, la pile d'adresse effectue une translation vers la droite.

La seule complication est la 9° disposition, les adresses de retour sont écrites dans un format différent que l'adresse pointée .

par exemple: si l'adresse de retour est "361Ø", elle devra être stockée dans la "pile" de retour comme "Ø76Ø" soit une forme compressée de "3610"

3610 = ØØ11 ØØØ1 Ø11Ø ØØØØ

pour les adresses programme , les 1°, 5°, 6° et 7° bits sont toujours à Ø si bien que la valeur maximum pour le premier digit est 7 (Ø111) et le second ØØØØ ou ØØØ1 Aussi dans les adresses deretour, les 3 bits utilisés sont"translatés"dans les 3 bits inutilisés du deuxième digit soit le 361Ø devient

Ø76Ø = ØØØØ Ø111 Ø11Ø ØØØØ

cette "compression" libère le premier digit qui contient d'autres informations. Si le premier digit est à ØØØØ, le retour est à une adresse programme utilisateur. S'il n'est pas à ØØØØ, le processeur sait que le retour est à l'adresse d'un périphérique. Par exemple ; pour l'imprimante, toutes les adresses débutent avec 6 = Ø11Ø.

Les fonctions synthétiques STO b ou RCL a permettent à l'utilisateur de déplacer le pointeur à n'importe quel endroit de la mémoire ainsi que les registres d'assignations ou les registres d'état. L'utilisation de "STO b" nous petmet de placer le pointeur directement à n'importe quel byte d'une ligne programme contenant plusieurs octets pour créer tout ce que vous voulez avec le BG (ou autres)

REGISTRES a
ET b

USER KEYS:
11 XROM 05,59
-11 XROM 01,59
12 XROM 05,60
-12 XROM 01,60
13 CODE
14 DECODE
-14 DEL
15 -PRINTER-
-15 PACK
-32 A

PRP ""

```
01♦LBL "?"
02 ENTER↑
03 ENTER↑
04 ENTER↑
05 ENTER↑
06 STOP
07 ENTER↑
08 ENTER↑
09 ENTER↑
10 ENTER↑
11 ENTER↑
12 ENTER↑
13 ENTER↑
14 STOP
15 ENTER↑
16 ENTER↑
17 ENTER↑
18 ENTER↑
19 ENTER↑
20 ENTER↑
21 ENTER↑
22 ENTER↑
23 ENTER↑
24 STOP
25 ENTER↑
26 ENTER↑
27 ENTER↑
28 ENTER↑
29 ENTER↑
30 ENTER↑
31 ENTER↑
32 ENTER↑
33 STOP
34 END
```

Au premier STOP, aprés avoir fait partir le programme "?" on éxécute un RCL b, que l'on décode. Le 4° digit avant la fin, est la position du pointeur programme dans le registre d'adresse absolue donnée par les 3 derniers digits de b.

Exemple 1:

b = ØØØØØØØØØØ4ØED

Au premier arrêt de la machine, le pointeur programme est sur le 4° octet du REG d'adresse absoluè ØED.

Exemple 2:

b = ØØØØØØØØØØ3ØEC

Au deuxième arrêt, sur le 3° octet du REG ØEC etc...

```
XEQ "?"
 RCL b
DECODE
000000000040ED
 RUN
 RCL b
DECODE
000000000030EC
 RUN
 RCL b
DECODE
000000000000EB
 RUN
 RCL b
DECODE
000000000050E9
```

ATTENTION! ceux qui veulent vérifier la position du pointeur à la main doivent tenir compte de la taille mémoire.

En SIZE n, RØØ est à l'ADR: 512 - n (en DEC).

* LE REGISTRE c

Il est de la forme: ABC ef 169 abc xyz

Les trois premiers digits ABC sont l'adresse absolue du 1° registre statistique et est spécifié par la fonction " REG". Cette adresse est changée à chaque utilisation de " REG", "SIZE" ou "PSIZE". Quand " +"; " -"; "CL "; "SDEV" ou "MEAN"sont éxécutées, le processeur se réfère aux digits choisis comme ABC pour connaître les registres utilisés comme registres de sommation

Les deux digits suivant "ef" sont utilisés comme mémoire de travail par l'imprimante

Les digits 6, 7 et 8 doivent garder une valeur constante de 169 sous peine de MEMORY LOST

Les digits abc et xyz correspondent respectivement aux adresses absolues du registre RØØ et du .END. final

quand vous utilisez le register c, vous pouvez ystocker n'importe quoi , même une valeur différente de 169 dans les digits 6, 7, 8, mais attention de ne jamais arrêter votre programme avant d'avoir réintégré cette valeur car à ce moment le processeur vérifie le registre c.

```
REGISTRE c

USER KEYS:
 11 XROM 01,61
 12 SIZE
 13 SIZE?
 14 DECODE
                        EREG  169  RØØ  END
            SIZE 000      2ØØ - RØØ = SIZE en HEXA
              RCL c       (512 dec = 2ØØ hex)
     -1.1002-11   ***     RØØ + ØØB =  REG position
             DECODE                    du 1° REG STAT.
                        20B  169  2ØØ  1FF
2080016920 01FE  ---->
            SIZE 001      2ØØ - 2ØØ = SIZE ØØØ
              RCL c       2ØØ + ØØB = R11 en 2ØB
     -1.0002-12   ***
             DECODE
                        20A  169  1FF  1FE
20A00169 1FF1FE  ---->
            SIZE 002      2ØØ - 1FF = SIZE ØØ1
              RCL c       1FF + ØØB = R11 en 2ØA
     -0.9002-13   ***
             DECODE
                        209  169  1FE  1FD
209001691FE1FD   ---->
            SIZE 100      2ØØ - 1FE = SIZE ØØ2
              RCL c       1FE + ØØB = R11 en 2Ø9
      "p=iORE"    ***
             DECODE
                        1A7  169  19C  19B
1A7001691 9C19B  ---->
            SIZE 200      2ØØ - 19C = SIZE Ø64
              RCL c                   en DEC 1ØØ
      "Ø=jA=7"    ***     19C + ØØB = R11 en 1A7
             DECODE
                        143  169  138  137
143001691 38137  ---->
            SIZE 300      2ØØ - 138 = SIZE ØC8
              RCL c                   en DEC 2ØØ
     =.?001+=3    ***     138 + ØØB = R11 en 143
             DECODE
                        0DF  169  0D4  0D3
0DF001690D40D3   ---->
            SIZE 317
              RCL c
     <.>001+<2    ***
             DECODE
                        0CF  169  0C3  0C2
0CF001690C30C2   ---->
```

Les trois derniers digits contiennent l'adresse absolue du dernier .END., qui est le premier maillon de la chaîne des LaBeLs et des END globaux. En effet, lors d'une instruction "GTO alpha","XEQ alpha" ou "CAT 1"; la machine commence la recherche au .END. final, car tous les LBLs globaux et les ENDs sont liés entre eux: chcun connaît l'adresse de celui qui le précède, et par conséquent, lors d'une recherche de LBL, la machine commence par la fin et remonte vers le premier LBL en sautant de END en LBL alpha. Ceci explique pourquoi, si 2 programmes ont le même nom, toute opération sur ce nom agira sur le dernier programme. D'autre part, lors d'un CAT 1, si la mémoire contient de nombreux programmes, les noms défilent lentement au début, puis de plus en plus vite, car la machine, àchaque nom, repart de la fin.

Voir dans les prochains JPC les programmes RCLR, STOR, SIZE? ... illustrant les propriétés du registre c: recherche synthetique du SIZE, stockage de n'importe quoi dans c stockage d'une valeur abhérante dans les registres STATistiques de c déplacement du rideau entre les registres de mémoire programme et de données.

00
01
02
03
04 UTILISATION
05
06 GENERALE
07
08
09
10
11 AUTO EXECUTION
12 DOUBLE LARGEUR
13 MINUSCULES
14 REECRITURE <SUR CARTE PROTOGEE>
15 IMPRIMANTE IL
16 IMPRIMANTE IL
17 ENREGISTREMENT INCOMPLET
18 UTILISATION GENERALE
19 EFFACES A
20 L'ALLUMAGE
21 VALIDATION DE L'IMPRIMANTE
22 ENTREE NUMERIQUE
23 ENTREE ALPHANUMERIQUE
24 DEPASSEMENTS DE CAPACITE IGNORES
25 ERREURS IGNOREES
26 VALIDATION DES TONES
27 MODE UTILISATEUR <USER>
28 VIRGULE/POINT DECIMAL
29 SEPARATION DES CHIFFRES PAR GROUPES DE 3
30 MODE CATALOGUE
31 MODULE HORLOGE DMY/MDY
32 ENTREE/SORTIE DU IL
33 MODULE IL
34 UTILISATION FUTURE
35 UTILISATION FUTURE
37 NOMBRE DE
37 DIGITS
38 APRES LA
39 VIRGULE
40 AFFICHAGE
41 AFFICHAGE
42 MODE TRIGONOMETRIQUE
43 MODE TRIGONOMETRIQUE
44 MODE ON CONTINU
45 ENTREE DONNEES
46 ENTREE ALPHA
47 MODE SHIFT
48 MODE ALPHA
49 BATERIE FAIBLE
50 MESSAGES
51 MODE SST
52 MODE PRGM
53 ENTREE/SORTIE
54 PAUSE <PSE>
55 EXISTANCE DE L'IMPRIMANTE

15	16	
0	0	MODE MAN
0	1	MODE NORM
1	0	MODE TRACE
1	1	MODE TR/STACK

36	37	38	39	
0	0	0	0	0 CHIFFRE
0	0	0	1	1 CHIFFRE
0	0	1	0	2 CHIFFRES
0	0	1	1	3 CHIFFRES
0	1	0	0	4 CHIFFRES
0	1	0	1	5 CHIFFRES
0	1	1	0	6 CHIFFRES
0	1	1	1	7 CHIFFRES
1	0	0	0	8 CHIFFRES
1	0	0	1	9 CHIFFRES
1	0	1	0	10 CHIFFRES
1	0	1	1	0 CHIFFRE
1	1	0	0	0 CHIFFRE
1	1	0	1	0 CHIFFRE
1	1	1	0	0 CHIFFRE
1	1	1	1	0 CHIFFRE

40	41	
0	0	MODE SCI
0	1	MODE ENG
1	0	MODE FIX
1	1	MODE FIX/ENG

42	43	
0	0	MODE DEG
0	1	MODE RAD
1	0	MODE GRAD
1	1	MODE RAD

× LE REGISTRE d

ou registre drapeau contient l'état binaire des 56 drapeaux de la machine. ce registre est fait de 56 bits dont chacun correspond exactement à 1 drapeau. le premier bit (ou le plus signifiant, ou le plus à gauche, ou le plus haut) correspond au drapéau ØØ et le cinquante sixième au drapeau 55

En chargeant le registre d, vous pouvez armer ou désarmer n'importe quel drapeau

Dans les prochains JPC voir les programmes TOGGLE FLAG (IF dans la PPC ROM); le TEXT ENABLER (V1N2), transformation HEX - OCT - HEX

suivant une idée de Francis Rozange, le programme suivant est un générateur automatique de buffer programme.

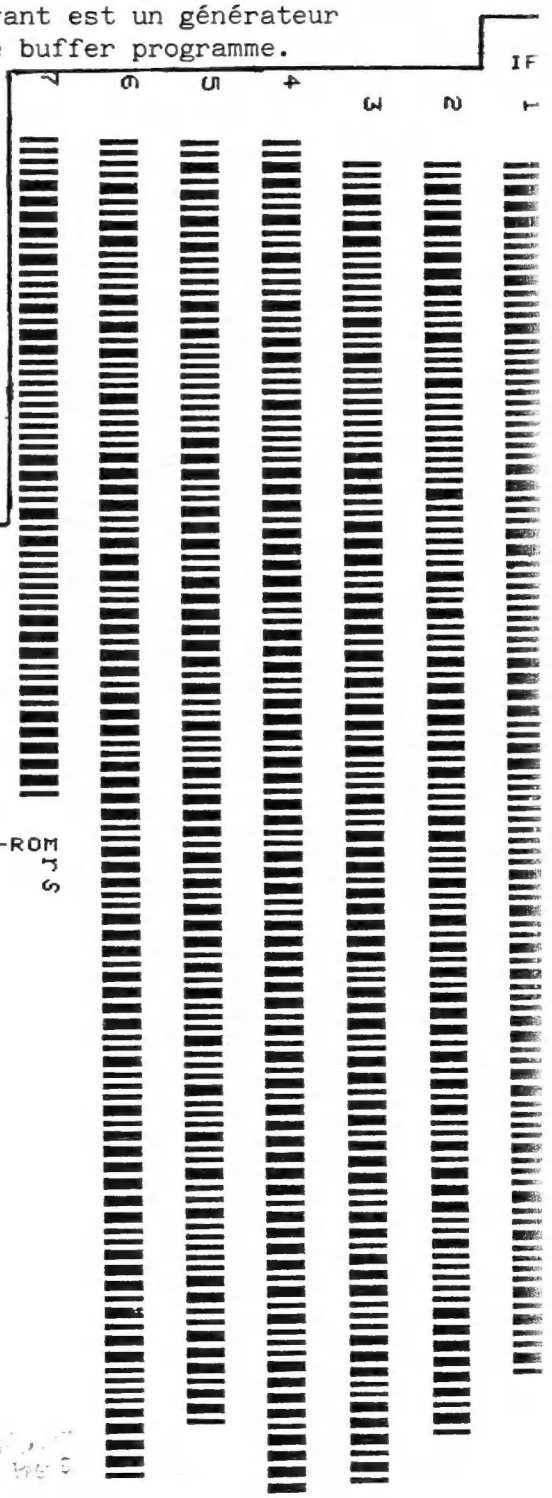
```
PRP "BUF"

01♦LBL "BUF"
02 52
03 TOGF (ou "IF")
04 51
05 TOGF (ou "IF")
06 END
```

```
01♦LBL "IF"
02 ABS
03 24
04 +
05 STO [
06 8
07 ST/ [
08 MOD
09 RCL d
10 X<> [
11 INT
12 SCI IND X (157,243)
13 ARCL X
14 X<>Y
15 X<> ]
16 X<> \
17 X<> d
18 FC?C IND ] (171,247)
19 SF IND ] (168,247)
20 X<> d
21 STO [
22 RDN
23 12
24 -
25 SCI IND X (157,243)
26 ARCL X
27 X<> ]
28 STO d
29 RDN
30 CLA
31 END (57 octets)
```

UTILISATION:
PLACER EN X LE NUMERO
DU DRAPEAU à INVERSER.
XEQ IF
EXEMPLE: IMP CONNECTÉE =
DRAP. 55 = LEVÉ
55 XEQ IF
DRAP. 55 = BAISSÉ =
CALCULS+ VITE ....

PPC-ROM
SJ

rs



```
255,000
     XEQ "DH"
FF'
        FIX 0
     82
     XEQ "DH"
52'
01♦LBL "DH"
02 CF 27
03 "                    "
04 RCL [
05 X<>Y

06♦LBL 03
07 ENTER↑
08 INT
09 16
10 MOD
11 9
12 -
13 X>0?
14 ISG X
15 ""
16 39
17 +
18 10↑X
19 STO ]
20 "├                   "
21 CLX
22 X<> ]
23 X<> \
24 STO [
25 RDN
26 16
27 /
28 INT
29 X≠0?
30 GTO 03

31♦LBL 05
32 "├ "
33 CLX
34 RCL [
35 X≠Y?
36 GTO 05
37 CLX
38 X<> ]
39 X<> \
40 STO [
41 PROMPT
42 END
```

CODES DEC:

P03: 254 39 32 32 32 32
32 32 32 32 32 32
32 32 32

P20: 247 127 32 32 32 32
32 32

P32: 242 127 32

LES PAS 09 ET 26 PEUVENT
ETRE REMPLACÉS PAR
RCL 06 .....(JPC N5 P27)

P15: 24Ø

RECTIFICATIONS

Une erreur s'est glissée P2 du N°5 de JPC où j'ai écrit à tort que les LBL ALPHA globaux ASN à des touches utilisaient 1 REG pour 2 ASN. Il n'en est rien et les ASN correspondantes sont STO "avec" ces LBL et ces ASN sont donc "gratuites" et enregistrées en mode USER. Je vous demande donc de m'excuser pour cette erreur.

J'en profite pour indiquer que le fameux "Buffer" est utilisé avec le nouveau module HP-IL DEVELopment (HP 00041-15043) que Jacques VAUCELLE vient de m'envoyer et qui est une petite merveille dans le genre. Je ne veux pas m'étendre ici sur les F (82) de ce module (toutes en MC) mais je vous indique simplement que la F 'BSIZEX' permet de créer ce fameux Tampon d'un nbre d'Octets précisé en x, que la F 'PT=' place le Pointeur sur l'Octet du Tampon dont le N° est précisé en x (semblable à 'SEEKPT'), que BSIZE? place en x la position du Pointeur, que PT? place en x la "dimension" (SIZE) du Tampon (excuses ! : inverser les 2 derniè res F pour obtenir les bonnes définitions), etc..., etc.... Indiquons tout de même les F 'HEXIN' (permet l'introduction d'un nbre HEX ainsi que sa conversion en DEC), HEXVIEW qui fait l'inverse de la précédente, BININ qui fait la même chose que HEXIN mais en BIN, BINVIEW qui....., 'XOR', 'AND', 'NOT', 'OR' et bien d'autres qui nous permettent notamment, de contrôler tout ce que "véhicule" la boucle "IL". Le manuel d'utilisation rédigé (comme toujours, en américain....) est de 78 pages et je ne me sens pas le courage, à l'heure actuelle, de vous résumer (?) les possibilités offertes par toutes ces nouvelles F. De plus je voudrais bien lire dans JPC la prose des autres car vous admettrez que pour l'instant, j'aurais tendance à crier "remboursez !".

Notre Président a élagué une partie importante de mon "papier" P4 du N°5 de JPC qui aurait dûe se trouver en marge et indiquer les CODes DEC à utiliser pour obtenir les fameuses (?) ASN. Je vais donc vous indiquer ces COD.

Touche 12: BG : 247 et la VAL de votre choix (8Ø, par exemple)

d°	22:	GTO "	: 4, 29 (ou 29,29)
d°	33:	LBL "	: 4, 2Ø5 (a __) (ou 2Ø5,2Ø5)
d°	64:	XEQ "	: 4, 3Ø (ou 3Ø,3Ø)
d°	63:	W "	: 4, 31 (ou 31,31)
d°	34:	mu = QL	: 4, 19 (ou 19,19)
d°	-34:	STO Q	: 145, 121
d°	43:	E	: 4, 27 (H __) (ou 27,27)
d°	51:	-	: 4, 28 (2 __) (ou 28,28)
d°	82:	,	: 4, 26 (ou 26,26)
d°	-41:	⁄ __	: 4, 160
d°	25:	μ __	: 4, 161
d°	-22:	eGOBEEP	: 4, 167 etc..., etc....

rS (P20)

CHOC EN RETOUR au V1N5P4 qui aurait du être publié avec l'article "ASN pour sport fr"

```
    CLKEYS
ASN "MKX" 83
   1 ENTER↑
  17 ENTER↑
 -11
  XEQ "MKX"
 247 ENTER↑
  63 ENTER↑
  12
  XEQ "MKX"
   1 ENTER↑
  16 ENTER↑
 -12
  XEQ "MKX"
  21 ENTER↑
  21 ENTER↑
 -13
  XEQ "MKX"
   1 ENTER↑
  20 ENTER↑
 -14
  XEQ "MKX"
   1 ENTER↑
  23 ENTER↑
 -21
  XEQ "MKX"
   1 ENTER↑
  29 ENTER↑
  22
  XEQ "MKX"
   0 ENTER↑
 167 ENTER↑
 -22
  XEQ "MKX"
   1 ENTER↑
  22 ENTER↑
  23
  XEQ "MK
   1 ENTER↑
  24 ENTER↑
 -23
  XEQ "MKX"
 205 ENTER↑
   0 ENTER↑
  33
  XEQ "MKX"
   1 ENTER↑
  19 ENTER↑
  34
  XEQ "MKX"
 145 ENTER↑
 121 ENTER↑
 -34
  XEQ "MKX"
   1 ENTER↑
  25 ENTER↑
 -42
  XEQ "MKX"
  27 ENTER↑
  27 ENTER↑
  43
  XEQ "MKX"
  28 ENTER↑
  28 ENTER↑
  51
  XEQ "MKX"
  31 ENTER↑
  31 ENTER↑
  63
  XEQ "MKX"
   1 ENTER↑
  30 ENTER↑
  64
  XEQ "MKX"

   1 ENTER↑
  21 ENTER↑
 -64
  XEQ "MKX"
   1 ENTER↑
  26 ENTER↑
  82
  XEQ "MKX"
ASN "PACK" 11
 ASN "CLP" 13
 ASN "DEL" 14
ASN "SIZE" 53
ASN "XTOA" 71
ASN "ATOX" 81
      PRKEYS
```

```
01 GTO "A"
02♦LBL "μ"
03♦LBL "A"
04  E
05 - E
06 123-
07 W "BEEP"
08 ;
09 "BST"
10 XEQ "×"
11 GTO "×α"
12 W "×αβ"
13 GTO "\J"&&"
14♦LBL "←"
15 "←X̄←P"
16 .END.
```

```
USER KEYS:
 11 PACK
-11 a×βdc'>52-$T+NμIAα ——— (1)
 12 XROM 28,63 ——————— ** BG **
-12 β ——————————— (0)
 13 CLP
-13 XROM "CJ" ——————— (5) XROM 20,21
 14 DEL
-14 DCAB@@←a&GB1HW> ——— (4)
-21 Aα ——————————— (7)
 22 @←AHHH ——————— GTO ...
-22 eG0BEEP ——————— XROM ...
 23 H_LD ——————————— (6)
-23 ≠αe2 ——————————— (8)
 33 XROM 52,00 ——————— LBL ...
 34 μΓ ——————————— (0L)
-34 XROM 05,57 ——————— STO 0
-42 OD ——————————— (9)
 43 XROM 44,27 ——————— E
 51 XROM 48,28 ——————— -
 53 SIZE
 63 XROM 60,31 ——————— W ...
 64 @NQ ——————————— XEQ ...
-64 X ——————————— (5)
 71 XTOA
 81 ATOX
 82 - ——————————— , (=0)
 83 "MKX"
```

```
*SUPER ASN * (P. S.)
3 PISTES/WSTS
```

Pool Sur

```
PRKEYS

USER KEYS:
 11 PACK
-11 a×βdc'>52-$T+NμIAα ——— (1)
 12 ACA ——————————— *** BG/LA
-12 β ——————————— (0)
 13 CLP
-13 XROM "CJ" ——————— (5) XR20,2
 14 DEL
-14 DCAB@@←a&GB1HW> ——— (4)
 15 XROM 01,53 ——————— RCL M
-15 XROM 05,53 ——————— STO M
 21 XROM 01,57 ——————— RCL Q
-21 Aα ——————————— (7)
 22 @←AHHH ——————— GTO ...
-22 eG0BEEP ——————— XR ...
 23 H_LD ——————————— (6)
 25 XROM 01,54 ——————— RCL N
-25 XROM 05,54 ——————— STO N
 32 XROM 01,62 ——————— RCL d
-32 XROM 05,62 ——————— STO d
 33 XROM 52,00 ——————— LBL ...
 34 μΓ ——————————— (0L)
-34 XROM 05,57 ——————— STO Q
 35 XROM 01,55 ——————— RCL O
-35 XROM 05,55 ——————— STO O
 43 XROM 44,27 ——————— E (=1)
 44 XROM 01,56 ——————— RCL P
-44 XROM 05,56 ——————— STO P
 51 XROM 48,28 ——————— - (CHS)
-51 XROM 01,61 ——————— RCL c
 53 SIZE
 61 XROM 01,60 ——————— RCL b
-61 XROM 05,60 ——————— STO b
 63 XROM 60,31 ——————— W ...
 64 @NQ ——————————— XEQ ...
 71 XTOA
 81 ATOX
 82 - ——————————— , (=0)
 83 XROM "1K" ——————— (PPC-ROM)
```

Aimé PIERARD
3, Bd de l'Yser
B 6000 CHARLEROI

Cher Ami,

Je ne peux résister à vous mettre un extrait d'une de mes lecture qu'il serait bon que les dévoués qui écrivent dans les revues spécialisées méditent, car il y a encore des personnes désireuses de n'initier aux nouvelles méthodes de calculs et qui s'en dégoûtent faute d'une bonne mise sur la voie.

Extrait de : La Nouvelle mathématique, entretien avec André Lichnerowitch, memebre de l'Institut de France, Edition Laffont.

Il existe de nombreuses manières de dire les choses : certaines sont très pittoresques comme "George Windsor, appelé roi d'Angleterre, est mort la nuit dernière" (Daily Worker, pages intérieures).

Pour beaucoup de mathématiciens de profession, probablement pour la plupart d'entre eux, la seule manière de les dire est cependant d'avoir recours à l'hermétisme. Une incursion à travers un texte mathématique n'est pas tâche aisée; habituellement, on n'y trouve pas le moindre exemple pratique. Tout appartient à ce que les ecolastiques auraient appelé le troisième degré de l'abstraction.

Quand un mathématicien veut expliquer les lois de Boole , il est rare qu'il prenne la peine de signaler que les circuits électriques en fournissent un modèle très clair. Il préfère l'élégante posture qui consiste à ne rien dire jusqu'à la fin. Le moins que l'on puisse dire de telles pratiques est qu'elles ne facilitent pas au public l'accès à la compréhension de la mathématique. Celle-ci lui fait l'effet d'une tour d'ivoire.

Remplacez mathématique par informatique et programmation synthétique car comme le disait encore PS dans un dernier n° HP est en dessous de tout pour les modes d'emploi de ses compléments.

Mais pour ma part j'ai essayé de me mettre à la P.S. et ne pouvant m'astreindre à un décriptage je ne suis parvenu à recopier aucun des programmes ne comportant ne fut-ce que quelques lignes. (synthétiques bien entendu)

Cependant je dis :Merci Robert (c'est de lui qu'il s'agit plus haut) car grâce à lui je possède le fameux XROM 28,63 et je dois à Philippe le XROM 29,01.

Mais pour ce qui est de reproduire les programmes RIEN.

J'ose espérer qu'après avoir lu ces lignes les futurs auteurs voudront bien redescendre au niveau où ils ont commencé et prendre les BEBES par la main avant qu'ils ne cassent leur beau Jouet.

Que les membres qui sont de mon avis veuillent bien le faire savoir au président afin d'avoir plus de "chaleur" au sein de notre grande famille.

Bonne programmation en HP.

MON CHER AIME,

J'ai parfaitement conscience que ta requête est justifiée, et je pense qu'à ce sujet tu n'as aucun soucis àte faire car le but de tout club de notre genre (PPC, PPCPC, PPCT, ...) est l'échange d'informations entre les divers membres. D'autre part, malgré que: "Tout ce qui ce comprend bien, s'énonce clairement", nous ne sommes pas des "professionnels" de l'éducation etmaquons quelquefois de clarté. Mais, c'est autres de nous poser des questions car il nous est impossible de deviner ce qui arrête chacun de vous et n'oublie

Wolfgang Baltes
Guardinistr. 143
D 8000 München 70

Chers amis,

je viens de recevoirla convocation à l'assemblée générale du PPC-Paris Chapter. Malheureusement je n'ai pas la possibilité d'aller à Paris, car je suis très occupé par beaucoup d'affaires, comme la préparation de l'assemblée générale du CCD et une action juridique contre l'ancien président du CCD (Rietschel) pour détournements de fonds....quelque chose, que je ne vous souhaite pas.

Une autre chose qui me préoccupe énormement est un projet du CCD appelé "CCD-ROM". C'est principalement pour cela que je vous écris. Vous allez trouver une petite présentation du CCD et de ce projet dans la lettre qui accompagne celle ci. Lisez la avant de continuer cette lettre.

Ici lire la lettre: chers Amis de ...
Philippe

Un projet comme le CCD-Rom demande beaucoup de travail! Pour minimiser les efforts inutiles il n'y a que trois personnes chargés d'écrire le microcode: Wilfried Kötz, Holger Adelmann et moi-même. Nous discutons lesquelles des fonctions proposées seront à écrire et comment le faire. Ensuite nous le faisons....(???!!!) Evidemment il y a plus d'idées que de possibilités de réalisation: Par exemple la fonction PASSWORD qui bloque l'accès à la machine sauf si on introduit le mot de passe préalablement stocké: cette protection doit être garantie même quand on enlève le module CCD de la calculatrice (en effacant tout son contenu, comme dans le cas de l'enlèvement des piles). Au début ceci semblait impossible à faire. Maintenant nous connaissons un chemin pour réaliser cette idée, mais ca ne sera pas facile..

D'ou tirons-nous ces idées? C'est par le journal du CCD, PRISMA, que nous recevons des suggestions. Après les premiers éssais de réalisation nous savons mieux ce qui sera réalisable et nous pouvons reparler aux membres du CCD. Bien entendu cette discussion ne concerne que l'aspect"utilisateur" de la fonction, la plupart des membres du CCD ne connaissant

pas la programmation en microcode.

Nous aimerions étendre cette discussion. Voici ma proposition: Le PPC-P installe un petit groupe "P-ROM", de deux ou trois personnes (peut être en commun avec le PPC-T) qui seront alors les responsables francais du projet CCD-ROM. Ces personnes doivent avoir beaucoup de temps libre à consacrer à ce travail.Elles seront chargées de verifier si les fonctions donnent les effets désirés (résultats ou messages d'érreur) en toute circonstance, et donner leurs idées concernant le manuel de la ROM. Pour pouvoir effectuer ce travail nous leur donnerons nos codes et tout le support nécessaire (un jeu d'EPROMs de 16 k avec des fonctions outil)(Nous ne pouvons fournir des MLDL ou semblable!). Les résultats de ces tests seront publiés dans les différents journaux des clubs. Nous pensons que cela reveillera l'intérêt général à ce projet considérablement. En plus cela stimulera la discussion sur la nécécité de telle ou telle fonction. Les personnes du groupe P-Rom auront de plus la possibilité de corriger et d'améliorer le code produit par nous ce qui sera profitable pour tout le monde. La commercialisation du module se fera evidemment également sous la responsabilité de ce groupe.

J'éspère que ces propositions vous plaisent, ou au moins vous servent de base pour me proposer quelque chose de mieux. Vous devez cependant comprendre qu'il sera nécessaire de limiter le nombre de personnes travaillant sur ce projet (directement) car une organisation efficace du projet ne serait pas possible autrement (déjà pour des raisons de coûts pour la communication entre ces gens!).

Finalement j'aimerais remarquer que HP-Allemagne nous fait subir le meilleur support possible pour ce projet.

C'est tout pour aujourd'hui. Passez une belle journée ensemble,

Wolfgang.

Wolfgang Baltes
Guardinistraße 143
D 8000 München 70

München, den 9.Juli 1983

Chers amis en France,

Je viens de recevoir les derniers numéros des chapters fran= cais de PPC de Toulouse et de Paris. Voyant que notre projet de ROM trouve beaucoup d'attention et sachant que le CCD n'est pas très bien connu en France, je vous écris ces lignes:

Qu'est ce le CCD ? Tout au début c'était un groupe d'élèves qui ont commencé à éditer une sorte de circulaire avec des programmes pour la HP-41. C'était il y a trois ans. Depuis, ce groupe a grandi, et il fallait l'organiser comme société enregistrée. Aujourd'hui le CCD (Com= puterclub Deutschland) a environ 1500 membres de toutes sortes de professions. Depuis le 28. mai il est présidé par Peter Kiefer (étudiant). Les autres membres du comité de direction sont Rolf Hansmann (chimiste, vice-prśident), Günter Lehrke (homme d'affaires, caissier), Andreas Marktscheffel (étudiant, 1. assesseur) et moi même (2. assesseur). Le journal - PRISMA - est édité par Rolf. La qualité d'impression provient de l'existence d'une machine à composer photographique qui produit une image de l'écriture et des graphiques sur un film (avec une résolution de 500 pixels/mm), à partir duquel on peut fabriquer les plaques d'impri= merie.

Contairement à nos amis étrangers en France et en Angleterre, nous ne sommes pas un chapter PPC. Nous avons une très bonne relation avec Hewlett-Packard, tout en gardant notre indépendance. Hp nous reconnaît comme une association de spécialistes de la HP-41 (si je peux m'exprimer ainsi) et nous aide en nous donnant autant d'information que possible. Il faut savoir que - le développement et la fabrication des calculatrices se trouvant aux USA - il n'y a que très peu d'information disponible en Europe. Nous essayons de montrer à HP qu'il y a un besoin pour cette in= formation, et nous avons l'impression que l'on veut satisfaire à notre demande.

Parlons du projet d'une ROM en microcode: Lors d'une présen= tation commerciale d'HP dans l'ancien opéra de Francfort (!!!), le CCD a obtenu la possibilité de se présenter aux clients d'HP. D'une façon totalement libre nous avons pu montrer la programmation synthétique, le langage machine, des modifications de hardware et des exemples d'appli= cations avec l'HP-IL. C'est là que nous avions parlé à plusieurs repré= sentants d'HP de Genève et des USA. On nous a promis du support pour un projet telque la fabrication d'un module ROM. Aussitôt nous avions pré= paré une réunion pour discuter ce projet, et ce n-était que quelques semaines plus tard que nous avions fait une demande officielle à HP. Pour de multiples raisons - échange des responsables chez HP à Corvallis, èlection du nouveau comité de direction du CCD, examens universitaires et bien d'autres encore - le projet n'a pas progressé beaucoup depuis. Nous sommes toujours en dialogue avec HP pour cela, et nous avons commencé à programmer les premières fonctions. Mais ça ne sera pas avant le mois d'octobre que cette partie du projet demarrera pleinement!

Nous pensons de faire participer nos amis à l'étranger à ce projet en publiant toute l'information sur ce qui a été fait et ce qui reste à faire. Aussi nous voulons que ce module sera disponible avec un manuel en français...

En espérant que ceci vous a donné quelques renseignements utiles, je vous souhaite une Heureuse Programmation,

Wolfgang

Wolfgang Baltes.

CHERS AMIS

Devant bientôt rencontrer nos amis allemand je vous demande de m'écrire pour connaître ceux qui pensent pouvoir faire parti du COROM (COmission pour l'etablissement de la ROM); 3 membres maximum seront choisis, mais tous les membres pourront y participer en envoyant leur programmes chez moi. Chacun de vous recevra un éxemplaire du questionnaire à remplir pour CHAQUE programmes soumis. Tout programme envoyé seul ou avec son questionnaire malrempli, sera refusé.
Chacun de vous connaitrons l'avancement de la CCD ROM dans des articles sous la rubrique "LA ROUTE DU ROM".
D'autre part j'attend vos idées et vos désirs des programmes et des routines que vous désirez trouver danscette ROM.

A TOUS UNE HEUREUSE PROGRAMMATION

PHILIPPE

REG M, N, O, P & ALPHA

Je lis avec beaucoup d'attention tous les articles publiés dans JPC (exceptés mes "papiers"!)

Page 2 du N°4, Philippe nous indique que, lorsque 24 CAR sont introduits en ALPHA, les premiers CAR sont poussés dans la partie haute de P. Nous allons voir ce que j'obtiens avec la "41CV" N° 2249S40690 :

```
0 SCI 9
    CLA
    STO ↑
ABCDEFGHIJK
ABCDEFGHIJKLMNOPQRSTUVWX
EFGHIJKLMNOPQRSTUVWXYZAB
    RCL ↑
-1,024344454-53   ***
    CLA
    STO [
QxCDEFG
    FIX 0
    ATOX
145,   ***
    ATOX
2,     ***
    SCI 9
    CLA
ABCDEFG
    RCL [
-1,424344454-53   ***
    CLA
    STO ↑
EFG◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
```

```
SCI 9 ____________________________ Ø,ØØØØØØØ ØØ
CLA STO P ALPHA ABCDEFGHIJK _____ ABCDEFGHIJK_
LMNOPQRSTUVWX  ("BEEP=TONE") _____ NOPQRSTUVWX_
YZAB ____________________________ RSTUVWXYZAB_
ALPHA (OFF) RCL P ________________ -1,Ø243444-53
CLA STO M ALPHA __________________ ■■CDEFG
```

Nous pouvons en déduire que seuls 2 CAR ont été poussés dans P et que les 2 Octets de G du REG P sont "réservés" à d'autres fins. Le décodage des novas (avec ATOX) me donne: 145 & 2. Il se trouvera peut-être quelqu'un pour nous expliquer ce qui s'est passé et à quoi correspondent ces 2 codes.

```
CLA ALPHA (ON) ABCDEFG ___________ ABCDEFG
ALPHA RCL M ______________________ -1,4243444-53
CLA STO P ________________________
```

Ceux qui n'ont pas d'IMP auront du mal à compter les NULs (barres supérieures). IL y en a 21 qui sont précédés de EFG et nous en déduirons que seulement 24 CAR sont "acceptés" en ALPHA comme nous l'indique le minable manuel d'utilisation de HP. Le REG P doit donc être utilisé avec beaucoup de circonspection.....

Philippe nous dit que ce REG (une partie) est utilisé en mode RUN, pour contrôler l'affichage et nous pouvons peut-être en déduire que c'est la cause des codes 145 & 2 découverts (?).

Je dis souvent que nous sommes encore bien loin d'être arrivés "AU FOND DE LA 41C".....

rS (P20)

SFA-RFA

Les heureux possesseurs du module X-FUNCTIONS ont la possibilité de créer des fichiers très pratiques:il s'agit des fichiers ASCII. Ce type de fichiers augmente considérablement les fonctions Alphanumériques de base d'une HP-4I, permettant de faire du "traitement de texte" ,à un faible niveau. Ces mêmes possesseurs seront sans-doute déçus,lors qu'ils liront dans la notice du XF que la sauvegar de de ces fichiers ne peut se faire que sur casset tes!Belle-mère HP semble en effet avoir oublié que l'achat du lecteur HP 82I6IA laisse un certain vi de dans un portefeuille...et qu'il existe un très bon lecteur de cartes magnétiques!

Comme les membres de PPC se veu lent des utilisateurs "pas comme les autres", ils ont entrepris de créer des programmes qui permet traient la sauvegarde de fichiers ASCII sur cartes magnétiques...et comme je suis un membre de PPC,il est facile de deviner que je m'y suis mis aussi... SFA(Sauvegarde de Fichier ASCII) et RFA(non,non,ce n'est pas pour flatter nos voisins Allemands de CCD,il s'agit de Rappel de Fichier ASCII!) sont le résultat d'une bonne quinzaine d'heures de travail avec ma HP-4ICV.Ces deux programmes sont pratiques et courts(ils tiennent sur une face de carte magné tique).Un avantage essentiel est qu'ils permettent d'économiser au maximum les cartes:un fichier de I6 registres(sans compter les deux registres d'en-tête) tiendra sur une face de carte.La méthode uti lisée permet d'exploiter à fond les sept octets de chaque registre:il n'y a pas de place perdue!Toute fois il est important de noter que l'on ne stocke pas le nom du fichier.
Comme toujours,ces avantages se paient, et il est indispensable de respecter certaines précautions d'emploi(pour l'enregistrement des fichiers) qui , si l'on regarde bien, ne sont pas si draconniennes.

MODE D'EMPLOI:

SFA:La grande contrainte est que le fichier ASCII que l'on désire sauvegarder doit être le pre mier des X-Mémoires.Comme la sauvegarde n'est justifiée que si le fichier est souvent utili sé(par prgm ou manuellement), il vous suffira de faire un effort pour l'occasion...
Une fois cette condition satisfaite, lancez le programme par XEQ RFA: à l'affichage du messa-ge "NOM?",vous introduisez le nom de ce même fichier(vous êtes en ALPHA).La HP ne le trouve ne pas toute seule,car le premier caractère du nom serait transformé par la normalisation...à moins que quelqu'un ne connaisse une méthode??
R/S:si le premier fichier n'est pas ASCII,vous aurez droit à un magnifique"FL TYPE ERR",sinon la HP vous demande les cartes et l'enregistre ment se fait.A la fin,votre fichier est intact.

```
01 LBL "SFA"
02 "NOM?"
03 AON
04 PROMPT
05 CLX
06 SEEKPTA
07 GETREC
08 I90
09 XEQ 00
10 AOFF
11 FLSIZE
12 PSIZE
13 GETR
14 WDTA
15 I90
16 GTO 00
17 LBL "RFA"
18 "Nb REG?"
19 PROMPT
20 PSIZE
21 "NOM?"
22 AON
23 PROMPT
24 AOFF
25 RDTA
26 CRFLD
27 SAVER
28 EMDIR
29 CLD
30 +
31 68
32 +
33 LBL 00
34 CLA
35 X() c
36 RCL IND c
37 STO M
38 ATOX
39 3
40 FS? 48
41 DSE X
42 *
43 XTOA
44 SIGN
45 CHS
46 AROT
47 X() M
48 STO IND c
49 R↑
50 STO c
51 CLA
52 END.
```

Lorsque vous enregistrez sur cartes un fichier ASCII,il est indispensable de noter sur la première carte la longueur de ce fichier en registres(nb qui apparaît lors d'un EMDIR) car elle ne sera pas enregistrée. Le nom a peu d'importance,vous pourrez donner au fichier le nom que vous voudrez lors de l'exécution de RFA. Une carte de sauvegarde d'un fichier d'adresse de 58 registres peut se présenter ainsi:

Adresses carte I
58 REGS

RFA:XEQ RFA:au message "Nb REG?" il vous faut introduire le nombre de registres occupés par le fichier(sans compter les deux registres d'en-tête). Ce nombre est sensé se trouver inscrit sur la première carte si vous avez suivi le mode d'emploi de SFA(?).
R/S:à l'affichage du message "NOM?",vous êtes en ALPHA et vous devez écrire le nom que vous désirez attribuer au fichier. Si vous introduisez plus de 7 caractères, seuls les 7 premiers seront pris.
R/S:La HP fait maintenant tout toute seule,ou presque,il vous suffit d'introduire les cartes demandées.Ne vous étonnez pas si elle exécute un EMDIR:je ne connais pas d'autre moyen pour déterminer la place restant en X-Mémoire.Le fichier est le dernier lors d'un EMDIR,de la même façon que lorsqu'on crée un fichier normal.

REMARQUE IMPORTANTE: Je ne possède pas les X-MEMOIRES;et le programme RFA nécessite quelques modifications selon que vous travaillez avec un ou deux X-MEMORY supplémentaires.Je n'ai pas cherché à effectuer ces modifications,mais j'indiquerai à quel endroit il faut les faire.

EFFETS:

La pile est perdue ainsi que ALPHA.
Le flag 48 est utilisé pour un branchement et est baissé à la fin des deux programmes.
Les programmes utilisent les registres R00 à Rnn avec:nn+I=Nb de registres du fichier.
SFA et RFA placent la HP en size(nn+I).

COMMENTAIRE DU PROGRAMME:

Le principe des programmes est assez simple en théorie:

Sauvegarde:On ne peut pas faire passer un fichier ASCII dans la MEV(la RAM) de la HP-41. Par contre, on peut le faire avec un fichier de données(DATA),puis le sauvegarder sur cassetes ou sur cartes.
Il "suffit" donc, avec un peu de PS, de transformer un fichier ASCII en fichier DATA!

On fait ensuite passer le fichier dans la MEV(on le recopie par GETR) et on retransforme le fichier en fichier ASCII. Comme il n'est pas possible(à ma connaissance) de déterminer l'adresse absolue d'un fichier situé en X-Mémoire, la seule solution est de travailler sur le premier fichier, dont l'adresse est fixe:ceci explique celà!
Toutefois,un gros problème se pose: les registres ne risquent-ils pas d'être normalisés au passage? On m'avait dit qu'un NNN(Nombre Non Normalisé) pouvait être stocké et rappelé sans problème de la mémoire d'extension.Par ailleurs, un NNN peut être stocké(mais pas rappelé) dans un registre de la 41. Enfin,le lecteur de cartes recopie les registres et les restitue sans normalisation (avec WDTA et RDTA!),il ne devait donc pas y avoir de problème...Toutefois, mes premiers essais furent des échecs:les fonctions GETRI et SAVERI normalisent en effet les registres!(ou bien une des deux seulement:je n'ai pas eu le courage de vérifier).Il en est peut-être de même avec les fonctions WDTAX et RDTAX(?).
Heureusement,les fonctions du XF: SAVER et GETR,ne posent pas de problème.....

Rappel: Là, on crée un fichier DATA à la taille désirée. Comme ce fichier est le dernier en X-Mémoires, on peut trouver son adres se et, après avoir récupéré les données, le transformer en ASCII: le tour est joué!

Pour ceux qui voudraient adapter ce programme à des configurations faisant intervenir des X-MEMORY, je donne un commentaire "ligne à ligne":

LIGNE:	COMMENTAIRE:
01 à 07:	Demande le nom du premier fichier et vérifie qu'il s'agit d'un fichier ASCII. (Celui-ci devient le fichier-travail).
08 à 09:	I90 est l'adresse absolue en décimal du deuxième registre d'en-tête du premier fichier. Le sous-programme 00 le transforme pour en faire un fichier DATA. Le flag 48 levé indique que l'on veut obtenir un fichier DATA, s'il était baissé, on obtiendrait un fichier ASCII.
10 à 14:	Rappel des registres dans la MEV et sauvegarde sur cartes.
15 à 16:	Le fichier redevient un fichier ASCII.
17 à 24:	Demande la taille du fichier et modifie la SIZE, demande aussi le nom du fichier que l'on rappelle. Ce nom peut être choisi en toute "liberté".
25 à 27:	Rappel des registres en MEV, puis création d'un fichier DATA et passage des registres dans le fichier.
28 à 32:	Calcul de l'adresse absolue du deuxième registre d'en-tête du fichier avant modification. Ce passage est à modifier si l'on travaille avec des X-MEMORY supplémentaires.
33 à 52:	Sous programme de modification des différents fichiers. On place le "rideau" en R(00) et on stocke en même temps l'adresse absolue du deuxième registre d'en-tête en c. L'en-tête est stocké en M, mais la normalisation a modifié le premier octet en mettant le premier digit à un. Les digits 2,3 et 4 (en partant de la gauche) contiennent l'adresse absolue du second registre d'en-tête. Ces trois digits sont à 0 si le fichier est vide (à la création). D'une part le fichier n'est pas vide, et d'autre part le digit 2 est à 0 car on travaille avec XF seul. La fonction ATOX donnera donc le chiffre I6. On multiplie ce nombre par deux ou 3, pour avoir un fichier DATA(32) ou bien ASCII(48). On replace l'octet en M(XTOA), puis on le replace en début de registre par AROT. On sait ici que le second octet n'était pas nul car on travaille sur XF (adresses de 040 à 0C0). Il semble que celà reste vrai pour les X-MEMORY(????) On stocke ensuite le registre d'en-tête modifié, et on rétablit c. Toute cette partie doit être modifiée si on utilise des X-MEMORY. J'ai essayé de grignoter les octets au maximum, celà risque donc de ne pas être facile. Quoi qu'il en soit, ne comptez pas trop y parvenir si vous ne connaissez pas le codage de l'entête des fichiers du XF(article de Jean FEVRE dans PPC-T N°6). Je reste à votre disposition pour toute information supplémentaire.....

Jean BOSCHAT(P5I+T7I)
4, rue Descartes,
59760 GRANDE-SYNTHE.

Lettre ouverte à Daniel JACOB

Bien cher collègue,

Je tiens à te remercier et à te féliciter très sincèrement pour ce que tu as publié dans JPC et je peux t'assurer que je suis un lecteur attentif et très intéressé par la technique des GTO et XEQ COMPilés. Ce qui a été écrit sur la question ne semble pas avoir été "AU FOND" du problème.....

Tes explications sont très claires et rédigées avec soin mais il serait peut-être utile, à l'attention des lecteurs "qui n'aiment pas trop chercher", de donner qq indications supplémentaires.

P16N5 de JPC, les 4 BITs 'r' codent le nbre ENTIER de REG et les 3 BITs 'o' codent le nbre d'Octets "restants" (ou MOD), c'est-à-dire : la "pesée" (°).

Avant COMPilation, c'est-à-dire avant exécution du PRGM, les 2ème et 3ème Octets des GTO (ou XEQ) sont à zéro.

Avec les GTO codés sur 2 Octets, ces 2 Octets sont pris en compte si le LBL correspondant se trouve en amont et ne sont pas comptés dans le cas contraire. (exemples : à 2 P16N5)

Avec les GTO & XEQ codés sur 3 Octets, seul le 1er Octet est pris en compte lorsque le LBL est situé en amont et dans le cas contraire, ce sont les 2 derniers Octets qui sont pris en compte. Tu as très bien indiqué cette particularité avec '1' et '2' placés respectivement sous et au dessus des accolades (crochets horizontaux). Cette petite précision permet d'affirmer qu'il n'y a aucune erreur P16 à 17 de JPC N°5 et permettra donc (?) à ceux qui veulent "COMPiler à la main", d'avoir les bases exactes pour le calcul du nbre d'Octets. Ils apprécieront d'autant plus le TRAVAIL que fait le PRGM "COMP" de FREDERIC, ce PRGM devant être suivi de l'exécution de ton PRGM "Pépé".....

J'ai tapé cette lettre au format '13,5cm' et je te joins une copie réduite (coef 0,7) afin que, si tu en juges l'opportunité, tu puisses l'envoyer pour publication dans JPC.

Je te renouvelle mes bien sinvères félicitations et je te prie de croire à ma profonde amitié,

rS (P20)

Mr PELANNE J-P
60 rue St.-Placide
75006 - PARIS

Cher Monsieur,

J'ai découvert de nombreuses fonctions intéressantes Les fonctions H-- et 2-- donnent respectivement E et - mais nécessitent pour cela une entrée numérique or j'ai trouvé les codes qui permettent d'obtenir E et - sans passer par H-- et 2--. E a pour code(27, 12) et - a pour code (28,12). On peut ainsi obtenir des lignes de 10, 15 ou 20 E et des -E très facilement, mais ce qui est encore plus intéressant c'est la possibilité de fabriquer des E1, E2, E3, E-1, E-2 etc..

En effet 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 etc, ont respectivement pour code :

0 : (16,12)	6 : (22,12)
1 : (17,12)	7 : (23,12)
2 : (18,12)	8 : (24,12)
3 : (19,12)	9 : (25,12)
4 : (20,12)	, : (26,12)
5 : (21,12)	

En ayant assigné par exemple E et - et 1 respectivement (27,12) , (28,12) et (17,12) on fabrique un E-1 en faisant en mode user prgm :

(01) E
(02) -
(03) 1

SST puis - puis BST et l'on voit :

(03) E-1
(04) "
(05) "

SST rétablit une numérotation normale des lignes.

Autre Ex :

Pour obtenir des E 100 ou E 1000, normalement impossible à obtenir on utilise la même méthode que ci dessus, mais on s'aperçoit qu'une fois qu'on a obtenu E 1000 si l'on passe en mode calcul et que l'on exécute la ligne, l'affichage nous donne 1,000000 10 il semblerait donc que les chiffres supplémentaires (deux zéros dans cet exemple) soient ignorés lorsque n dépasse 9,999999 99

La fonction X52,00 sert à fabriquer des labels délocalisés mais il existe une autre fonction qui permet elle aussi de les créer c'est la fonction ■ -- dont le code est (14,205)
La fonction A- code (14,23) est intéressante, prenons un exemple :
Faisons en mode prgm user

	A A	→	7	(01)
	A B	→	7	(02)
	A C	→	7	(03)
SST		→	ᵀC	(04)
SST		→	ᵀB	(05)
SST		→	ᵀA	(06)
SST	SST		777	(01)
			ᵀC	(02)
			ᵀB	(03)
			ᵀA	(04)

La chose intéressante c'est que le paramètre rentré est stocké en ordre inverse (cf mon exemple) par fabrication d'une chaine contenant celui-ci.

Pour finir j'évoquerais le cas de la fonction P c non programmable mais assignable à une touche. P c → GTO .. en mode prgm seulement dans le cas où le prgm n'est pas private sinon P c est inactif.

Happy programming
J.F. PELANNE P48

P.S.

Pour répondre à la question de Robert Schwartz à propos du mot NYBBLE, je dirais que nous savons que les chiffres 0 à 9 sont codés sur 4 bits en binaire par 0000 à 1001. Une unité est constitué de 4 bits [illegible] l'appelle NYBBLE ou " digit ". Ces 4 bits [illegible]urraient être appelés éléments d'informations, en [illegible]fet prenons un exemple, soit le nombre :

2,537148107 E 45

Ce nombre sera codé comme 14 NYBBLES (14x4 bits) de la forme :

0000 0010 0101 0011 0111 0001 0100 1000 0001 0000
\+ 2 5 3 7 1 4 8 1 0

0111 0000 0100 0101
7 + 4 5

ou bien en 7 bits de la forme :

02 53 71 48 10 70 45

mais ceux ci représentent les mêmes bits que dans un prgm c'est à dire les lignes de prgm suivantes :

LBL01 , Y↑x , x<>y , Σ⁻ , 0 , CLΣ & x>y ?

Les 7 bits ci-dessus correspondent aux codes HEX et les instructions de prgm à leur valeur dans le tableau des codes.

COURRIER DU COEUR

A vendre module HPIL neuf (600FF). Serenseigner auprés de DAMIEN DEBRIL, 38 rue du 8 Mai 1945, 59190 HAZEBROUCK
Recherche lecteur de cartes magnétiques à bas prix. Toutes propositions à ERIC DELAGNES, 14 rue Merlin De Thionville, 92150 SURESNES.
Recherche QUADRAM àbas prix. Toutes propositions à ERIC MONSENEGO,105 rue De La Convention, 75015 PARIS.

S.O.S.

FRANCK LEBASTARD Cherche comment suprimer une alarme déterminée ou toutes à la fois par mode programme.
Est-il possible de placer par programme n'importe quelle instruction à un endroit déterminé de la mémoire, et ceci sans utiliser de zone tampon

PIERRE LAMY Si un dés membres du PPCPC peut programmer un logiciel de comptabilité avec la HP41C/CV et avec une relance automatique des retardataires et informations au secretariat pour la tenue du ou des fichiers.

AIME PIERARD Pourquoi abandonnez-vous les"conventionnels", il y a notamment certains programmes qui tourneraient trés bien en notation normale.
De plus, il me paraît opportun de créer une rubrique comme dans KEY-NOTES : ROUTINES ET TOUR DE MAIN par exemple

DAMIEN DEBRIL Y a-t-il d'autres membres du club qui ont récupérés la HP75?
J'ai eu l'occasion de voir la ROM VISICALC, c'est super !!!

ERIC MONSENEGO Cherche qui pourrait l'aider à placer un interupteur pour l'EPROM du HLDL permettant son adressage ou on le désire.

URGENT * URGENT * URGENT * URGENT * URGENT * URGENT * URGENT

Pour cause de changement d'emploi, je vends toute mon installation professionnelle, système HP . Ci-dessous le détail de ce que je possède. Tout est en excellent état de marche et est fourni avec toute la documentation nécessaire. Malheureusement, je ne peux pas vendre ce matériel séparément. Je m'adresse donc plutôt a un intermédiaire qui achèterait mon matériel "CASH", et le revendrait par la suite comme il l'entend. En effet, j'ai besoin urgent de ce capital, car je dois rénover une salle de sport. Je suis prêt à apporter ce matériel ,moi-même, à la personne intéressée. La HP-41 C que je possède a été modifiée par M. Jacques Vaucelle et comporte 2 jeux de 3 modules groupés (1 X-Fonctions + 2 X-Memoire; au total 2 X-Fonctions et 4 X-Mémoire). De plus, un de ces deux groupes est à l'intérieur de la machine ainsi que un Module Time + 4 Modules Mémoire , qui sont aussi soudés dans le ventre de la HP. Le chargeur et la batterie sont aussi modifiés et permettent une recharge plus rapide. Ce matériel ne m'a jamais causé d'ennuis.

LISTE DE MATERIEL

INTERFACE VIDEO 82163 B
LECTEUR DE CASSETTES 82161 A
IMPRIMANTE IL 82162 A
MODULE IL 82160 A
MODULE PPCROM
MODULE MATH 14023
3 CHARGEURS
11 CASSETTES
2 LECTEURS DE CODES BARRE (un n'a jamais servi;je l'ai reçu en cadeau) 82153 A
1 PORT EXTENDER
4 BOITES DE PAPIER THERMIQUE (NOIR ET BLEU)
1 LECTEUR DE CARTES MAGNETIQUES 82104 A
240 CARTES MAGNETIQUES
UNE CAISSE (1m. x 50 cm. x 50 cm.) DE LIVRES ,DOCUMENTATION ET PROGRAMMES
UNE HP-41C AVEC A L'INTERIEUR :
- 1 MODULE TIME 82182 A
- 4 MODULES MEMOIRE
- 1 X-Fonctions + 2 X-Mémoire groupés
- batterie modifiée 6V

(DANS LE PORT No. 1 EST INSERE UN DEUXIEME GROUPE DE 1 X-Fonctions + 2 X-Mémoire)
EN PLUS JE REMETS AUSSI UNE AUTRE HP-41C EN PARFAIT ETAT DE MARCHE MAIS SANS BOITIER INFERIEUR

* * * * * * * * * * * *

JE CEDE LE TOUT POUR 5000.- FRANCS SUISSES
(ENVIRON 18 000.- FRANCS FRANCAIS)

MARCO BENEDETTI
RUE STAEMPFLI 132
2503- BIENNE
SUISSE Tél. 032/ 41 92 51

PPCPC (PARIS CHAPTER) No. 29
PPCT (TOULOUSE) No. 65

M. Benedetti

```
10 ! TELephone
20 T=.60 ! Taxe de base
30 INPUT "DUREE : ";D
40 INPUT "PLEIN TARIF=1 : DEMI TARIF=2 : ";C
50 D=D*C @ DISP"PRET" @ GOSUB 90
60 ON TIMER # 1,D P=P+T @ GOSUB 100
70 P=T @ GOSUB 100 @ GOSUB 90
80 OFF TIMER # 1 @ END
90 IF KEY$="" Then 90 ELSE RETURN
100 DISP "Prix =";P @ RETURN
```

Commentaires:

A la ligne 20 vous modifiez le cöefficient T suivant le rythme de l'inflation .
Ligne 30 D indique la durée correspondant à la taxe de base.
Ligne 40 D est multiplié par 2 lors du demi tarif.
Ligne 60 intervient le chronomètre n° 1 qui ajoute la valeur de T au prix précédent, ceci chaque fois que la durée D s'est écoulée.
Ligne 70 Premier affichage du prix.
Ligne 80 Arrêt du chronomètre n° 1.
Ligne 90 Boucle utilisée lorsqu'aucune touche n'est frappée.
Ligne 100 Sous programme d'affichage du prix.

Toutes mes excuses pour les fautes de frappe.

Sur le HP75C il ya autant de chronomètres que le permet la taille mémoire. Ce nombre est limité à 1001 chronos en tout!!!

Damien DEBRIL
38, Rue du 8 Mai 1945
59190 HAZEBROUCK

PPCPC 22

A la ligne 20 vous modi-
-fiez le coefficient T
suivant le rythme de
l'inflation.
Ligne 30 D indique la
duree correspondant a
la taxe de base.
Ligne 40 D est multiplie
par 2 lors du demi-tarif
Ligne 60 intervient le
chronometre # 1 qui
ajoute la valeur de T au
prix precedent, ceci cha
que fois que la duree D
s'est ecoulee.
Ligne 70 Premier affiche
-ge du prix.
Ligne 80 Arret du chrono
#1.
Ligne 90 Boucle utilisee
lorsqu'aucune touche
n'est frappee.
Ligne 100 Sous-program-
-me d'affichage du prix.

Sur le HP75C il y a au-
-tant de chronos que le
permet la taille memoire
Ce nombre est limite a
1001 chronos en tout !.

```
10 ! TELephone
20 T=.6 ! Taxe de base
30 INPUT "DUREE : ";D
40 INPUT "PL. TARIF=1 *D
EMI TARIF=2 :";C
50 D=D*C # DISP "PRET" @
 GOSUB 90
60 ON TIMER # 1,D P=P+T
@ GOSUB 100
70 P=T # GOSUB 100 # GOS
UB 90
80 OFF TIMER # 1 @ END
90 IF KEY$="" THEN 90 EL
SE RETURN
100 DISP "Prix =";P # RE
TURN
```

PC # 22

PRGM sur K7
++++++++++

Je me tiens à la disposition des collègues du Club qui désirent une copie des PRGM publiés dans JPC pour effectuer ces enregistrements.

M'envoyer une K7 initialisée (NEWM) dans un emballage solide(une petite boîte en fer placée dans une enveloppe "à coussins d'air" peut convenir, pdr exemple). Notre ami Damien DEBRIL vous confirmera qu'on ne prend jamais assez de précautions.... J'en profite pour lui demander de nous écrire un article sur la répartition des FICHiers sur une K7 (c'est un expert en la matière).

Prévoir l'enveloppe affranchie pour le retour (self-adressée) et ne pas faire d'expéditions [redacted] recommandé car cela m'obligerait à aller à la Poste (on peut me faire confiance, je ne mange pas de ce pain là!).

M'indiquer les PRGM désirés qui seront VERIFY et SEC.

Robert SCHWARTZ
141, avenue Carnot
93140 BONDY

Recensement des BUGs

Je pense qu'il serait intéressant que tous ceux qui ont découvert dans des bouquins ou à la suite de recherches personnelles, des Fonctions "particulières" telles que les nombreux BG, BJ, -PRINTER-, mu, QL, etc..., y compris les BUGs qui semblent, à priori, inutilisables (?), de nous envoyer les CODES XROM ainsi que les CODES DEC (pour ASN) correspondants.

Je vais donc commencer la liste :

CODE XROM	CODE/ASN	Origine (?)
28,63	247,63	PPC-US (Keith JARETT)
3Ø,14	247,142	PPC-US (John DEARING)
eGOBEEP	Ø,167	PPC-US (Keith JARETT)
μ (mu)	4,19	(XR 12,19) Jean THIBERGE
μ --	Ø,16Ø	(XRØØ à63) Jean THIBERGE(+rS)
μ --	Ø,161	(XRØ5 -35) Jean THIBERGE(+rS)
44,ØØ	27,Ø	QL (Keith JARETT)
29,Ø1		BG (Lionel ANCELET)
52,29	4,29	GTO "délocalisés"(voir "rS")
52,ØØ	2Ø5,2Ø5	LBL d° (d°)
44,27	27;27	E (d°)
48,28	28,28	- (d°)
60,31	31,31	W " (d°)
56,3Ø	3Ø,3Ø	XEQ d° (d°)

Nous les accepterons tous pour un inventaire, même les "ratons-laveurs" (Prévert).....

D'avance, MERCI rS (P20)

Ø5,Ø3	161,67	"Cric" Philippe DESCAMPS,
Ø5,Ø3	241,67	" d° " Patrick IMBAULT
		Bruno LANGLOIS

Excusez-moi pour cet oubli qui aurait été impardonnable.

Pour les F de Jean THIBERGE, se reporter à l'OP N°6 P53.

Dernière minute: Je viens de découvrir tout à fait par hazard, un nouveau BG : XROM 29,13 dont les CODES DEC pour ASN sont 247,77 et qui s'affiche : "M[redacted]" Je vais peut-être finir par en trouver un dont la lettre sera P comme PPC(?).

Robert, tu patines dans la choucroute !

J'ai écrit qu'il n'était pas possible de rentrer [redacted] ALPHA plus de 12 CAR identiques et notre charmant collègue, Bruno TREDEZ, m'a gentiment montré, lors de la dernière réunion chez Jacques VAUCELLE, qu'il était parfaitement possible de "continuer" et que je m'étais simplement fait trahir par [redacted] simple effet d'optique.....

Donc, merci Bruno, je vais pouvoir BBBBégayer plus de 12 fois..... rS (P20)

Bizarre, vous avez dit bizarre ?

Utilisez-donc les CODES Ø & 254 pour ASN la F[redacted] et voyons ce que nous pouvons en faire.

Mode PRGM GTO .. LBL "T" ENTER ENTER ENTER ENTER ENTER GTO . ØØ1 XEQ "PACK"

Nous allons admettre que la F Ø,254 est ASN à la touche 25 et j'indiquerai donc TAN à chaque fois que cette F sera à utiliser. Signalons que cette F nous invite à introduire 2 chiffres directs ou INDirects et que les ENTER n'ont été placés que par mesure de sécurité afin de ne pas nous retrouver dans le domaine d'Alice.

(suite) "ABCDEFG" BST PACK TAN 15 (bizarre ?) des SST vous permettront de contrôler la présence des ENTER dont le nbre doit être "copieux", GTO .ØØ2 ← si vous avez placé suffisamment d'ENTER, SST vous fera découvrir ceux qui restent, Retapez un bon paquet d'ENTER,PACK puis effacez 12 ENTER (GTO .Ø13 et 12 fois ←)

TAN 99 (bizarre ?) Si vous n'aviez pas effacé les 12 ENTER, vous auriez trouvé à la place des NULS (barres sup), des représentations de CAR ayant le code 131 qui est....ENTER et ce petit "c", d'où sort'il ? : du 99 que vous avez tapé et vous pourrez donc placer comme second CAR de la chaîne,tous les CAR compris entre 15 (tiens, tiens!) et 99 et je vous laisse découvrir ce que donnent les VAL inf à 15.

L'utilisation de l'INDirection ne m'a rien apporté de significatif mais vous pouvez tout de même essayer. Il est évident que nous pouvons remplacer les 12 premiers ENTER par d'autres F et vous trouverez peut-être une application à ce BUG bizarre, vous avez dit bizarre ? rS (P20)

Quelques précisions [redacted] "HN"

Le PRGM "HN" précédemment publié, convertit [redacted] expression HEXadécimale (14 DIGits au maxi) placée en ALPHA, [redacted] NNN (Nombre Non Noramlisé) placé [redacted] X et la CHaîne ALPHA correspondante est placée [redacted] ALPHA.

Not'Sident nous indique P26 de JPC N°2, que la CH obtenue avec "HN" peut être STOckée dans un REG "normal" tel que RØØ par exemple.

OUI MAIS, dans un REG "normal", nous [redacted] pouvons STOcker (ASTO rr) que 6 CAR [redacted] maximum et [redacted] expression HEX de 13 ou 14 DIG [redacted] placera 7 CAR en ALPHA. Il [redacted] faudra donc veiller à contrôler cette limitation [redacted] moyen de ALENG, si [redacted] "travaillons" dans un PRGM et "visuellement", si nous opérons [redacted] mode RUN.

Il y [redacted] hélas une autre limitation. Nous [redacted] que nous pouvons placer en ALPHA 24 CAR [redacted] maximum mais, si en mode RUN vous essayez de rentrer 24 "F", par exemple, vous constaterez qu'il n'est pas possible d'en rentrer plus de 12 et même APPEND (SHIFT XEQ) est inopérant. C'est le système de COMPilation qui nous interdit d'entrer plus de 12 CAR identiques et vous remarquerez que cette limitation s'applique aussi bien aux CAR ALPHA qu'aux chiffres ("111111111111").

Je laisse à un collègue dévoué et "qui sait", le soin de nous expliquer le système de COMPilation d'une CHaîne ALPHA et not'Sident nous [redacted] donné quelques indications à ce sujet (JPC N°2P26).

Lorsque [redacted] stockons (ASTO rr) [redacted] CH ALPHA (6 CAR max) dans un REG "normal", cette CH est "identifiée" par le préfixe F suivi du nbre de CAR de la CH et les 2 DIG correspondants occupent l'Octet "qui nous manque".et pour éviter d'utiliser l'HEX, [redacted] pouvons dire que l'Octet d'identification d'une CH ALPHA est DEC 24Ø + le nbre de CAR (24Ø+6=246, au maximum).

Philippe nous [redacted] fait judicieusement remarquer qu'il est indispensable, après avoir rappelé [redacted] ALPHA la CH désirée (CLA ARCL rr), d'utiliser RCL M pour placer [redacted] X le NNN correspondant qui élimine l'Octet (No byte, please! que nous pourrions confondre [redacted] BIT) d'identification.

Signalons [redacted] passage, qu'un NNN n'est pas "normalisé" (modifié) tant qu'il est maintenu dans la PILE ou (et) dans le REG L (LASTx) et les "sportifs" utiliseront [redacted] priorité cette particularité.....

Je pense qu'il était utile de donner [redacted] quelques précisions et j'indique à [redacted] collègues qu'ils peuvent [redacted] écrire pour [redacted] préciser quels sujets particuliers ils aimeraient [redacted] voir développer. Les TEN, Q LOADER (mu [redacted] l'OI) et eJUMP ("masque"?) me paraissent très intéressants à étudier et je crois que Philippe devra [redacted] donner des précisions supplémentaires [redacted] permettant de bien assimiler le "POURQUOI et le COMMENT". rS (P20)

```
 4:43AM 23.06
01♦LBL "SK"
02 SIGN
03 CLX
04 X<> "
05 XEQ 14
06 ISG L
07 ""
08 .
09 X<> e
10♦LBL 14
11 "*"
12 X<> [
13 STO \
14 ASTO IND L
15 RDN
16 RTN
17♦LBL "RK"
18 SIGN
19 ARCL IND L
20 "⊢♦"
21 ISG L
22 ""
23 ARCL IND L
24 "⊢‡"
25 X<> \
26 STO '
27 X<> [
28 STO e
29 RDN
30 CLA
31 RTN
32 END
```

PRKEYS

```
USER KEYS:
 11 PACK
 12 XROM 28,63
 13 CLP
 14 DEL
 21 XROM 08,18
 22 XROM 52,29
 23 XROM 24,22
 24 XROM 01,53
-24 XROM 05,53
 25 XROM 01,57
-25 XROM 05,57
 33 XROM 52,00
 34 XROM 12,19
 43 XROM 44,27
 51 XROM 48,28
 53 SIZE
-53 XROM 10,15
 61 BST
 62 XROM 16,20
 63 XROM 60,31
-63 #ae2
 64 XROM 56,30
 71 XROM 20,21
 72 μΓ
 73 XROM 32,24
 81 ATOX
-81 XTOA
```

Suspendre et Régénérer les ASN

Un coup de téléphone d'un collègue me demande quelques renseignements dont un relatif à la suppression temporaire des ASN.

Bien que ce problème soit du domaine de la PS, donc en dehors de celui pour lequel je m'étais engagé à vous aider, je me suis rappelé qu'il existait dans la PPC-ROM, 2 PRGM qui devaient faire l'affaire.

Voici donc les PRGM "SK" & "RK" écrits par Keith JARETT en réponse à une suggestion de Gary M. TENZER (sic);

LBL "SK"

Ø2 SIGN moyen élégant de stocker en L le premier REG spécifié en X

Ø3 CLX place Ø en X

Ø4 X<>⊢ rappelle en X le NNN correspondant au REG interne qui contient l'index des touches ASN et place Ø dans ce REG

Ø5 XEQ 14 exécute la sous-routine 14 qui stocke les Octets (4+ 4BITs) adéquats dans le REG spécifié

Ø6 ISG L incrémente L de 1 pour définir le 2ème REG

Ø7 "" Fonction inopérante destinée à "neutraliser" le "test" précédent

Ø8 . place Ø en X

Ø9 X<> e rappelle en X le NNN du REG interne qui contient les index d'ASN des touches SHIFTées et place Ø dans ce REG e

1Ø LBL 14 s/routine de stockage

11 "*" efface ALPHA et y place une étoile (sic), CAR CODE 42 dans le 1er Octet (de G à D) du REG interne M

12 X<>M le NNN correspondant au contenu du REG M crée au Pas précédent est placé en X et le NNN récupéré au Pas Ø9 est placé en M

13 STO N le nouveau contenu d'ALPHA (l'étoile + les 7 Octets (7 premiers de G à D) du REG cité au Pas Ø9 sont stockés en N

14 ASTO IND L stocke dans le REG spécifié, l'étoile + les 5 premiers Octets du REG cité au Pas Ø9

15 RDN décale la PILE d'1 cran vers le Bas (remet la PILE en place)

16 RTN

Il est peut-être bon de rappeler que le REG interne ⊢ contient l'index des touches ASN qui est réparti dans les 9 premiers DIGits (de G à D). Cet index est en quelque sorte le "répertoire" des touches ASN et il suffit donc d'y placer Ø pour faire croire à notre machine qu'il n'y a pas d'ASN. Les Fonctions et (ou) les LBL ALPHA globaux sont eux, stockés dans les REG "normaux" situés juste au-dessus du RIDeau, c'est-à-dire à partir de l'ADDresse DEC 192.

Le REGe contient l'index des touches SHIFTées qui sont ASN.

```
SIZE 002
0,
XEQ "SK"
RCL 00
"*"00?1"    ***
VIEW 01
"* #"
PRKEYS

USER KEYS:NONE
0,
XEQ "RK"
PRKEYS

USER KEYS:
 11 PACK
 12 XROM 28,63
 13 CLP
 14 DEL
 21 XROM 08,18
 22 XROM 52,29
 23 XROM 24,22
 24 XROM 01,53
-24 XROM 05,53
 25 XROM 01,57
-25 XROM 05,57
 33 XROM 52,00
 34 XROM 12,19
 43 XROM 44,27
 51 XROM 48,28
 53 SIZE
-53 XROM 10,15
 61 BST
 62 XROM 16,20
 63 XROM 60,31
-63 *ee2
 64 XROM 56,30
 71 XROM 20,21
 72 µΓ
 73 XROM 32,24
 81 ATOX
-81 XTOA
```

```
17 LBL "RK"
18 SIGN      place en L le N° du REG spécifié en X
19 ARCL IND  L rappelle en ALPHA le contenu du REG
             spécifié au Pas précédent
2Ø "├◆"      place un Octet NUL à la D de ALPHA
21 ISG L     incrémente L pour obtenir l'ADD du
             second REG
22 ""        même utilité qu'au Pas Ø7
23 ARCL IND  L rappelle en ALPHA le contenu du
             second REG
24 "├Φ"      ajoute à la D de ALPHA les CODEs DEC
             15 & 255 (HEX ØF FF)
25 X<> N     place en X le NNN correspondant aux
             premiers 5 Octets du REG├ + le NUL
             + l'étoile
26 STO├      remet en place le "répertoire" index
27 X<> M     place en X le NNN correspondant aux
             5 premiers Octets du REG e + ØF FF
28 STO e     rétablit les index d'ASN des touches
             SHIFTées
29 RDN       remet la PILE en place
3Ø CLA       efface ALPHA
31 RTN
```

Pour pouvoir suivre ce qui se passe dans le REG ALPHA, je vous conseille de faire un croquis comme celui indiqué dans JPC N°2 P10.

Il convient également de bien visualiser le contenu des REG internes e & ├ .

Vous avez donc remarqué (?) qu'il suffit de 2 REG "normaux" pour stocker le "répertoire" des touches ASN.

<u>UTILISATION</u> : Placer en X l'ADD du 1er REG dans lequel sera stocké le ½ répertoire après avoir fait le SIZE adéquat, puis XEQ "SK". Reportez-vous à la bande d'essais ci-jointe en remarquant qu'il est parfaitement possible d'ASN des Fonctions ou (et) des LBL entre "SK" et "RK" (au moyen de PASN, par exemple). Le contenu des 2 REG spécifiés peut être enregistré sur une PISTe de CARTe magnétique (WDTAX).

Un collègue plus qualifié pourra nous donner des renseignements plus précis sur les ASN.

rS (P20)

NNN = Nbre Non Normalisé
ADD = Adresse

```
XROM "A?"
14,    ***
```

CODES DEC : PØ7 & 22: 24Ø
P2Ø: 242 127 Ø
P24: 243 127 15 255 (le CODE 255 n'est pas IMPrimé car il est réservé pour empêcher l'avance du papier avec la touche PAPER ADVANCE.

LE PETIT THEATRE DES MICROCODES

CODES RETOURNES PAR C = KEY:

A la pression d'une touche, le drapeau KB est levé et un code HEXadécimal est charge dans le registre KEY (digits 3 - 4). si aucune touche n'est pressée aprés l'exécution de C=KEY, ØØ est chargé. Tous les autres digits de C, ne sont pas affectés
Le programme "KEYS" ci-contre est celui qui ma permis de retrouver tous les codes de touche de la 41

```
F600 093 "S"
F601 019 "Y"
F602 005 "E"
F603 00B "K"
F604 345
F605 040 ?NCXQ 10D1 ---> (CLA)
F606 2DC R= 13
F607 150 LDØR 5 }
F608 110 LDØR 4 }      T
F609 110 LDØR 4 }
F60A 3D0 LDØR F }      O
F60B 150 LDØR 5 }
F60C 150 LDØR 5 }      U
F60D 110 LDØR 4 }
F60E 0D0 LDØR 3 }      C
F60F 110 LDØR 4 }
F610 210 LDØR 8 }      H
F611 110 LDØR 4 }
F612 150 LDØR 5 }      E
F613 0D0 LDØR 3 }
F614 3D0 LDØR F }      ?
F615 168 WRIT 5(M)
F616 191
F617 00C ?NCXQ 0364 ---> (AVIEW)
F618 3C8 CLRKEY <----
F619 3CC ?KEY        |
F61A 3F3 JNC F618 -02 ---
F61B 04E C=0 ALL
F61C 220 C=KEY KY
F61D 3B5
F61E 051 ?CXQ 14ED ---> (RDN)
F61F 0E8 WRIT 3(X)
F620 381
F621 040 ?NCXQ 10E0 ---> (CLD)
F622 3C8 CLRKEY <----
F623 3CC ?KEY        |
F624 3F7 JC F622 -02 ---
F625 0E8 WRIT 3(X)
F626 3E0 RTN
```

ce tableau represente les touches du calculateur:

18	C6	C5	C4
10	30	70	80	C0
11	31	71	81	C1
12	32	72	82	C2
13		73	83	C3
14	34	74	84
15	35	75	85
16	36	76	86
17	37	77	87

On fait partir la routine KEYS:
TOUCHE? est affichée;
on presse la clé de notre choix
puis on decode le contenu de X qui de la forme:
ØØ ØØ ØØ ØØ Øx xØ ØØ
ou xx est le code de la touche pressée

```
                    KEYS
TOUCHE?  (9)
                  DECODE
00000000004000
                    KEYS
TOUCHE?  (6)
                  DECODE
00000000005000
                    KEYS
TOUCHE?  (ALPHA)
                  DECODE
00000000004000
```

NOM D'UNE FONCTION, PROMPT, ...

La 41 reconnait deux listes de caractères: le ASCII modifié (voir la table HEXadécimale) etla liste de caractères en ROM. Les 2 points (3A) sont affichés comme un "pâté"; la virgule (2C) et le point (2E)donnent respectivement à l'affichage le"canard" droit et gauche quand il sont utilisés dans un nom de fonction ou à l'affichage.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	↑	_
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	├	a	b	c	d	e	f	'	Γ	σ	β	:	μ	≠	Σ	∡

Quand une fonction est éxécutée, l'opérating système controle le contenu du premier mot ROM (les deux premiers caractères) du nom de la fonction ainsi que les deux mot ROM suivant. L'entrée du catalogue pour une fonction microcodée pointe le code du premier mot éxécutable (voir JPC V1N6). Par exemple l'entrée de la fonction CLA (HEXA 87) qui a l'entrée du cat. en 1487 dont le contenu du mot ROM est Ø1D soit l'adresse du premier mot ROM éxécutable.

```
1486 0BB
1487 0D1 → adresse de la fonction CLA
1488 092
```

Le listing de la fonction CLA montre que : quand la fonction est en microcode, son nom est écrit en sens inverse.
le premier mot ROM éxecutable (en 1ØD1) est non nul donc la fonction est programmable.
le dernier caractère du nom de la fonction est identifié en ajoutant Ø8Ø ici à l'adresse 1ØCE rien n'a été ajouté au mot ROM du premier caractère, donc aucun arrêt particulier.

```
10CE 081 "A"
10CF 00C "L"
10D0 003 "C"
10D1 04E C=0 ALL
10D2 168 WRIT 5(M)
10D3 1A8 WRIT 6(N)
10D4 1E8 WRIT 7(O)
10D5 228 WRIT 8(P)
10D6 3E0 RTN
```

Le système opératoire examine d'abord le contenu des mots correspondant au deux premiers caractères de la fonction et éxécute l'arrêt demandé (voir tableau ci dessous). Si l'arrêt accepte une ligne ALPHAnumérique, celle ci est stockée dans le registre Q (en HEXA), justifiée à droite et en sens contraire de la lecture. Par exemple: ASN "COPY" charge en Q ØØ ØØ ØØ 59 5Ø 4F 4C, avant l'éxécution de la routine ASN. Si l'arrêt est numérique, la donnée est chargée dans le registre interne A en binaire.
Une entrée numérique 55; retourne ØØ ØØ ØØ ØØ ØØ ØØ 37 en A . Et ajoute 8Ø HEX pour un IND; l'entrée de IND 55 retourne ØØ ØØ ØØ ØØ ØØ ØØ B7 en A

```
TAN
0,0000    ***
RCL _
0,0000-46 ***
DECODE
00000000 4E 154
T A N
```

```
F700 0B1 "I"
F701 009 "I"
F702 001 "A"
F703 013 "S"
F704 113 "S"
F705 205 "E"
F706 0AE A<>C ALL
F707 0E8 WRIT 3(X)
F708 3E0 RTN
```

```
ASN "ESSAI1" 1
ASN "DECODE" 1
ASN "PRA" 1
ESSAI1 5
DECOD
F8FFFFF000101 37  (55)10 = (37)16
ESSAI1 IND 5
DECOD
F8FFFFF000101 37  B7=80+3
ESSAI1
DECOD
F8FFFFF000101 3C  (60)10 = (3C)16
ESSAI1 IND 1
DECOD
F8FFFFF000101 3C  3C=
```

EXEMPLE	ADDITIONNE EN HEXA AU 1'CAR	2'CAR	TYPE D'ARRET DEMANDE:
CLA	000	(TOUT)	AUCUN
CLP	100	000	DEMANDE ENTREE ALPHA (NUL REFUSEE)
SIZE	100	100	DEMANDE 3 DIGITS (4 SI PRESSION SUR EEX)
	100	200	DEMANDE ENTREE ALPHA NON NULLE
CAT	100	300	DEMANDE 1 DIGIT OU IND OU IND ST
STO	200	000	DEMANDE 2 DIGITS OU IND OU IND ST OU ST OU OPERATIONS (-,+,*,/)
RCL	200	100	DEMANDE 2 DIGITS OU IND OU IND ST OU ST
FS?	200	200	DEMANDE 2 DIGITS OU IND OU IND ST
	200	300	DEMANDE 2 DIGITS OU IND OU IND ST
LBL	300	000	DEMANDE NON NUL ALPHA OU [illegible]
XEQ	300	100	DEMANDE NON NUL ALPHA OU 2 DIGITS OU IND OU IND ST
	300	200	DEMANDE NON NUL ALPHA OU 2 DIGITS
GTO	300	300	DEMANDE NON NUL ALPHA OU 2 DIGITS OU IND OU IND ST OU [illegible]

Aprés les deux premiers caractères du nom de la fonction, deux autres mots sont examinés par le système opératoire. Ils sont lus UNIQUEMENT en mode PRGM mais TOTALEMENT ignorés en mode RUN. Ce sont les deux premiers mots éxécutables soient ceux qui suivent directement le nom de la fonction.

Si le premier mot éxécutable est ØØØ; la fonction est alors non programmable.

Si les deux premiers mots éxécutables d'une fonction microcodée sont ensembles ØØØ; la fonction est alors non programmable et immediatement éxécutable (on l'appellera: NPIE).
Si vous pressez une touche à laquelle on a assigné une NPIE et que vous êtes en mode PRGM, cette fonction est éxécutée répététivement jusqu'au moment ou vous relachez la touche.

EXEMPLES DE FONCTIONS.

```
10C5 094 "T"
10C6 301 "A"
10C7 103 "C"
10C8 201
10C9 02E ?NCGO 0B80
10CA 286 C=0-C S&X
```

```
1203 094 "T"
1204 010 "P"
1205 00D "M"
1206 00F "O"
1207 012 "R"
1208 010 "P"
1209 281
120A 00E ?NCGO 03A0
```

```
122B 08C "L"
122C 103 "C"
122D 212 "R"
122E 18C ?FSET 11
122F 3B5
1230 051 ?CXQ 14ED
1231 04E C=0 ALL
```

```
129B 094 "T"
129C 013 "S"
129D 013 "S"
129E 000 NOP
129F 181
12A0 08A ?NCGO 2260
```

```
128E 085 "E"
128F 01A "Z"
1290 109 "I"
1291 113 "S"
1292 255
1293 05E ?NCGO 1795
```

EXEMPLE DE NPIE

```
E800 085 "E"
E801 009 "I"
E802 010 "P"
E803 00E "N"
E804 000 NOP
E805 000 NOP
E806 151
E807 14C ?NCXQ 5354 (RUNSW)
E808 368 WRIT 13(c)
E809 149
E80A 024 ?NCXQ 0952
E80B 0F8 READ 3(X)
E80C 2A0 SETDEC
E80D 10E A=C ALL
E80E 04E C=0 ALL
E80F 22E C=C+1 ALL
E810 23C RCR 2
E811 01D
E812 060 ?NCXQ 1807
E813 0E8 WRIT 3(X)
E814 2D5 (PRX)
E815 1BC ?NCXQ 6FB5
E816 3C8 CLRKEY
E817 3CC ?KEY
E818 360 ?C RTN
E819 261
E81A 000 ?NCXQ 0098
E81B 151
E81C 14C ?NCXQ 5354 STOPSW)
E81D 328 WRIT 12(b)
E81E 065 (RCLSW)
E81F 144 ?NCXQ 5119
E820 3E0 RTN
```

Incremente le reg x

TESTE SI UNE TOUCHE EST PRESSEE

```
0952 046 C=0 S&X
0953 3F0 PRPH SLCT
0954 270 RAM SLCT
0955 3E0 RTN
```

La deuxième fonction (voir page 24 bis) 0098 est une sous routine HP permettant le test de pression d'une touche par analyse du drapeau KEY et chargement du registre KEY

La troisième routine (TEMPS) sert a décoder le registre x apres rappel de la fonction du TIME: RCLSW.

Sous la forme xxxx h xx mn xx s, xx

La pression sur la touche assignée à Σ+ appelée NPIE donne l'impression sur l'imprimante des 2 bandes ci dessous.

```
          ASN "NPIE" 11
          ASN "TEMPS" 12
                 GTO ..
PACKING
                    CLA
                    CLX
          0,000     ***
                  FIX 0
              1     ***
              2     ***
              3     ***
              4     ***
              5     ***
              6     ***
              7     ***
              8     ***
              9     ***
             10     ***
             11     ***
             12     ***
          3,-04     ***
                  TEMPS
0h 00mn 02s ,78
```

```
                    CLX
          0         ***
                    CLA
          1         ***
          2         ***
          3         ***
          4         ***
          5         ***
         10         ***
         11         ***
         12         ***
         13         ***
         14         ***
         15         ***
         16         ***
         17         ***
         18         ***
         19         ***
         20         ***
       6,-04        ***
                  TEMPS
0h 00mn 06s ,44
```

La troisième routine (HP) (voir page 24 ter) sert à faire une somme entre les 2 registres C et A. Somme normalisée pour chargement dans X.
?NCX 1807 fait C = C+A ALL, resultat en C.

Teste une pression de touche

```
0098 3C8 CLRKEY
0099 3CC ?KEY
009A 3F7 JC 0098 -02
009B 130 LDI S&X
009C 008
009D 3C8 CLRKEY
009E 3CC ?KEY
009F 266 C=C-1 S&X
00A0 3EB JNC 009D -03
00A1 3E0 RTN
```

```
E701 093 "S"
E702 010 "P"
E703 00D "M"
E704 005 "E"
E705 014 "T"
E706 2A0 SETDEC
E707 04E C=0 ALL
E708 270 RAM SLCT
E709 0F8 READ 3(X)
E70A 10E A=C ALL
E70B 08E B=A ALL
E70C 130 LDI S&X
E70D 003
E70E 246 C=A-C S&X
E70F 2E6 ?C≠0 S&X
E710 08B JNC E721 +11
E711 0A0 SLCT P
E712 39C R= 0
E713 10E A=C ALL
E714 03C RCR 3
E715 0AE A<>C ALL
E716 11A A=C M
E717 11E A=C MS
E718 0AE A<>C ALL
E719 33C RCR 1
E71A 0AE A<>C ALL
E71B 106 A=C S&X
E71C 0AE A<>C ALL
E71D 2E2 ?C≠0 ØR
E71E 013 JNC E720 +02
E71F 222 C=C+1 ØR
E720 38B JNC E711 -0F
E721 00E A=0 ALL
E722 02E B=0 ALL
E723 0A0 SLCT P
E724 25C R= 9
E725 0E0 SLCT Q
E726 35C R= 12
E727 112 A=C P-Q
E728 05C R= 4
E729 0A0 SLCT P
E72A 01C R= 3
E72B 0F2 C<>B P-Q
E72C 09C R= 5
E72D 0E0 SLCT Q
E72E 15C R= 6
E72F 0F2 C<>B P-Q
E730 04A C=0 R<
E731 27C RCR 9
E732 04A C=0 R<
E733 0BE A<>C MS
E734 33C RCR 1
E735 0BE A<>C MS
E736 0BC RCR 5
E737 158 M=C ALL
E738 0AE A<>C ALL
E739 1BC RCR 11
E73A 0BE A<>C MS
E73B 0A6 A<>C S&X
E73C 33C RCR 1
E73D 0BE A<>C MS
E73E 33C RCR 1
E73F 0A6 A<>C S&X
E740 33C RCR 1
E741 0A6 A<>C S&X
E742 33C RCR 1
E743 0A6 A<>C S&X
E744 13C RCR 8
E745 0EE C<>B ALL
E746 0BC RCR 5
E747 0BE A<>C MS
E748 33C RCR 1
E749 0BE A<>C MS
E74A 33C RCR 1
E74B 0BE A<>C MS
E74C 23C RCR 2
E74D 0BE A<>C MS
E74E 13C RCR 8
E74F 0A6 A<>C S&X
E750 2BC RCR 7
E751 0A6 A<>C S&X
E752 0AE A<>C ALL
E753 0A0 SLCT P
E754 2DC R= 13
E755 0D0 LDØR 3
E756 010 LDØR 0
E757 0D0 LDØR 3
E758 010 LDØR Ø
E759 1D0 LDØR 7 } s
E75A 0D0 LDØR 3
E75B 090 LDØR 2 } espace
E75C 010 LDØR 0
E75D 090 LDØR 2 } ,
E75E 310 LDØR C
E75F 0D0 LDØR 3
E760 010 LDØR Ø
E761 0D0 LDØR 3
E762 260 SETHEX
E763 20E C=C+A ALL
E764 168 WRIT 5(M)
E765 04E C=0 ALL
E766 29C R= 7
E767 0D0 LDØR 3
E768 010 LDØR 0
E769 0D0 LDØR 3
E76A 010 LDØR 0
E76B 0D0 LDØR 3
E76C 010 LDØR Ø
E76D 0D0 LDØR 3
E76E 06E A<>B ALL
E76F 20E C=C+A ALL
E770 15C R= 6
E771 2E2 ?C≠0 ØR
E772 04F JC E77B +09
E773 0E0 SLCT Q
E774 2DC R= 13
E775 052 C=0 P-Q
E776 0A0 SLCT P
E777 3D4 R=R-1
E778 3D4 R=R-1
E779 394 ?R= Ø
E77A 3BB JNC E771 -09
E77B 1E8 WRIT 7(O)
E77C 198 C=M ALL
E77D 00E A=0 ALL
E77E 0AE A<>C ALL
E77F 2DC R= 13
E780 190 LDØR 6 } h
E781 210 LDØR 8
E782 090 LDØR 2 } espace
E783 010 LDØR Ø
E784 0D0 LDØR 3
E785 010 LDØR Ø
E786 0D0 LDØR 3
E787 010 LDØR 0
E788 190 LDØR 6 } m
E789 350 LDØR D
E78A 190 LDØR 6 } m
E78B 390 LDØR E
E78C 090 LDØR 2 → espace
E78D 20E C=C+A ALL
E78E 1A8 WRIT 6(N)
E78F 3E0 RTN
E790 090 LDØR 2
E791 20E C=C+A ALL
E792 1A8 WRIT 6(N)
E793 261 } ?NCXQ
E794 188 ?NCXQ 6298 } PRA
E795 3E0 RTN
```

Ce programme Temps que j'ai fait specialement pour la NPIE me permet de normaliser le chromomètre sous la forme h, mn, s, –

```
ASN "RUNSW" 11
ASN "RCLSW" 12
ASN "TEMPS" 13

FIX 9
RUNSW
RCLSW
0.003264000   ***
TEMPS
0h 00mn 32s ,64
RCLSW
0.130109000   ***
TEMPS
0h 13mn 01s ,09
```

JPC
VOL. 1 N7
PAGE 24 bis

```
1807 02E B=0 ALL
1808 07A A<>B M
1809 1D8 C<>M ALL
180A 198 C=M ALL
180B 046 C=0 S&X
180C 03E B=0 MS
180D 05E C=0 MS
180E 1D8 C<>M ALL
180F 31C R= 1
1810 3D4 R=R-1
1811 176 A=A+1 XS
1812 236 C=C+1 XS
1813 354 ?R= 12
1814 3E3 JNC 1810 -04
1815 0EE C<>B ALL
1816 2EE ?C≠0 ALL
1817 16B JNC 1844 +2D
1818 1D8 C<>M ALL
1819 06E A<>B ALL
181A 2EE ?C≠0 ALL
181B 14B JNC 1844 +29
181C 326 ?A<B S&X
181D 087 JC 182D +10
181E 1D8 C<>M ALL
181F 06E A<>B ALL
1820 326 ?A<B S&X
1821 3DF JC 181C -05
1822 0DE A<>C ALL
1823 1D8 " "
1824 36E "N"
1825 023 JNC 1829 +04
1826 1D8 C<>M ALL
1827 0AE A<>C ALL
1828 02D JNC 182D +05
1829 1D8 C<>M ALL
182A 0AE A<>C ALL
182B 1D8 C<>M ALL
182C 06E A<>B ALL
182D 19E A=A-B MS
182E 35E ?A≠0 MS
182F 05B JNC 183A +0B
1830 28E C=0-C ALL
1831 11E A=C MS
1832 326 ?A<B S&X
1833 03B JNC 183A +07
1834 1D8 C<>M ALL
1835 2FC RCR 13
1836 1D8 C<>M ALL
1837 06E A<>B ALL
1838 1A6 A=A-1 S&X
1839 06E A<>B ALL
183A 326 ?A<B S&X
183B 04D JNC 1844 +09
183C 166 A=A+1 S&X
183D 3CE RSHFC ALL
183E 0BE A<>C MS
183F 11E A=C MS
1840 3D4 R=R-1
1841 2D4 ?R= 13
1842 3C3 JNC 183A -08
1843 04E C=0 ALL
1844 0AE A<>C ALL
1845 198 C=M ALL
1846 0EE C<>B ALL
1847 12E A=A+B ALL
1848 276 C=C-1 XS
1849 276 C=C-1 XS
184A 276 C=C-1 XS
184B 000 NOP
184C 02E JNC 1855 +19
184D 02E B=0 ALL
184E 07A A<>B M
184F 1D8 C<>M ALL
1850 198 C=M ALL
1851 046 C=0 S&X
1852 03E B=0 MS
1853 05E C=0 MS
1854 1D8 C<>M ALL
1855 206 C=C+A S&X
1856 25E C=A-C MS
1857 013 JNC 1859 +02
1858 29E C=0-C MS
1859 00E A=0 ALL
185A 1D8 C<>M ALL
185B 2DC R= 13
185C 3DC R=R+1
185D 38E RSHFA ALL
185E 013 JNC 1860 +02
185F 12E A=A+B ALL
1860 262 C=C-1 @R
1861 3F3 JNC 185F -02
1862 354 ?R= 12
1863 3CB JNC 185C -07
1864 198 C=M ALL
1865 0AE A<>C ALL
1866 1D8 C<>M ALL
1867 198 C=M ALL
1868 2FE ?C≠0 MS
1869 01B JNC 186C +03
186A 166 A=A+1 S&X
186B 3CE RSHFC ALL
186C 0AE A<>C ALL
186D 35C R= 12
186E 34A ?A≠0 R<
186F 00F JC 1886 +17
1870 0EE C<>B ALL
1871 0AE A<>C ALL
1872 35C R= 12
1873 10E A=C ALL
1874 1E6 C=C+C S&X
1875 06B JNC 1882 +0D
1876 23A C=C+1 M
1877 05D JNC 1882 +0B
1878 0C6 C=B S&X
1879 226 C=C+1 S&X
187A 222 C=C+1 @R
187B 0DE C=B MS
187C 06E A<>B ALL
187D 2FA ?C≠0 M
187E 360 ?C RTN
187F 04E C=0 ALL
1880 01E A=0 MS
1881 3E0 RTN
1882 0C6 C=B S&X
1883 3C3 JNC 187B -08
1884 06E A<>B ALL
1885 363 JNC 1871 -14
1886 342 ?A≠0 MR
1887 34F JC 1870 -17
1888 266 C=C-1 S&X
1889 3EA LSHFA R<
188A 3E3 JNC 1886 -04
188B 02E B=0 ALL
188C 0FA C<>B M
188D 0AE A<>C ALL
188E 04E C=0 ALL
188F 1D8 C<>M ALL
1890 04E C=0 ALL
1891 35C R= 12
1892 05A LDPP 1
1893 0EE C<>B ALL
1894 0AE A<>C ALL
1895 1D8 C<>M ALL
1896 0AE A<>C ALL
1897 033 JNC 189D +06
1898 02E B=0 ALL
1899 07A A<>B M
189A 1D8 C<>M ALL
189B 198 C=M ALL
189C 046 C=0 S&X
189D 03E B=0 MS
189E 2FA ?C≠0 M
189F 123 JNC 18C3 +24
18A0 1D8 C<>M ALL
18A1 246 C=A-C S&X
18A2 25E C=A-C MS
18A3 013 JNC 18A5 +02
18A4 29E C=0-C MS
18A5 1D8 C<>M ALL
18A6 05E C=0 MS
18A7 0AE A<>C ALL
18A8 06E A<>B ALL
18A9 32E ?A<B ALL
18AA 02B JNC 18AF +05
18AB 1D8 C<>M ALL
18AC 3EE LSHFA ALL
18AD 266 C=C-1 S&X
18AE 1D6 " "
18AF 35C " "
18B0 04E C=0 ALL
18B1 013 JNC 18B3 +02
18B2 222 C=C+1 @R
18B3 18E A=A-B ALL
18B4 3F3 JNC 18B2 -02
18B5 12E A=A+B ALL
18B6 3EE LSHFA ALL
18B7 3D4 R=R-1
18B8 2D4 ?R= 13
18B9 3D3 JNC 18B3 -06
18BA 0AE A<>C ALL
18BB 198 C=M ALL
18BC 1C1
18BD 062 ?NCGO 1870
18BE 02E B=0 ALL
18BF 0FA C<>B M
18C0 0AE A<>C ALL
18C1 35E ?A≠0 MS
18C2 01B JNC 18C5 +03
18C3 0B5
18C4 0A2 ?NCGO 282D
18C5 03E B=0 MS
18C6 0CE "Σ"
18C7 06E " "
18C8 1EE " "
18C9 1EE C=C+C ALL
18CA 20E C=C+A ALL
18CB 0EE C<>B ALL
18CC 05A C=0 M
18CD 10E A=C ALL
18CE 1EE C=C+C ALL
18CF 1E6 C=C+C S&X
18D0 013 JNC 18D2 +02
18D1 20A C=-C-1 M
18D2 14E A=A+C ALL
18D3 39C R= A
18D4 342 ?A≠0 MR
18D5 017 JC 18D7 +02
18D6 3AE RSHFB ALL
18D7 38E RSHFA ALL
18D8 04E C=0 ALL
18D9 2DC R= 13
18DA 150 LDPP 5
18DB 3CE RSHFC ALL
18DC 053 JNC 18E7 +0A
18DD 222 C=C+1 @R
18DE 1CE A=A-C ALL
18DF 3F3 JNC 18DE -02
18E0 14E A=A+C ALL
18E1 3EE LSHFA ALL
18E2 394 ?R= 0
18E3 1C9
18E4 063 ?CGO 1872
18E5 3D4 R=R-1
18E6 3CA RSHFC R<
18E7 3BB JNC 18DF -09
18E8 1A6 A=A-1 S&X
18E9 03B JNC 18F1 +07
18EA 2C3 JNC 18C3 -28
18EB 2FE ?C≠0 MS
18EC 2B7 JC 18C3 -2A
18ED 2F6 ?C≠0 XS
18EE 2A7 JC 18C3 -2C
18EF 10E A=C ALL
18F0 0BE B=A ALL
18F1 01C R= 3
18F2 3EE LSHFA ALL
18F3 38A RSHFA R<
18F4 35C R= 12
18F5 34A ?A≠0 R<
18F6 397 JC 18E9 -0E
18F7 166 A=A+1 S&X
18F8 3A6 ?A<C S&X
18F9 01B JNC 18FD +03
18FA 236 C=C+1 XS
18FB 3E0 RTN
18FC 04E C=0 ALL
18FD 222 C=C+1 @R
18FE 3CE RSHFC ALL
18FF 23E C=C+1 MS
1900 0EE C<>B ALL
1901 2C2 ?B≠0 @R
1902 01B JNC 1906 +03
1903 3AA RSHFB R<
1904 226 C=C+1 S&X
1905 00E A=0 ALL
1906 1C2 A=A-C @R
1907 023 JNC 190C +04
1908 3EE LSHFA ALL
1909 12E A=A+B ALL
190A 3FB JNC 190A -01
190B 1BE A=A-C MS
190C 033 JNC 1913 +06
190D 38A RSHFA R<
190E 16E A=A+1 ALL
190F 226 C=C+1 S&X
1910 12E A=A+B ALL
1911 3FB JNC 1911 -01
1912 0AA A<>B R<
1913 262 C=C-1 MR
1914 36B JNC 1902 -13
1915 27E C=C-1 MS
1916 35B JNC 1902 -15
1917 3EE LSHFA ALL
1918 0A6 B=A S&X
1919 05A C=0 M
191A 05E C=0 MS
191B 12A A=A+B R<
191C 14E A=A+C ALL
191D 0BA A<>C M
191E 3E0 RTN
```

Gentils Membres

Je suis un peu déçu de la réaction de certains au sujet des dessins dans le J.P.C. . Effectivement, j'ai eu des échos d'amis qui ont reçu des coups de téléphone et provoquant une véritable polémique à ce sujet .

Si certains pensent que "philosopher" est plus important que notre passion à tous -la programmation- dans ce cas, d'accord, je suis prêt à changer les objectifs du club. Mais croyez-moi, ces dessins ne sont fait sans arrière pensée et éxistent pour la joie de tous de lire un journal illustré et avec une petite pointe d'humour. Je crois que nous sommes généralement trop sérieux dans la vie, et qu'aujourd'hui le rire a du mal à subsister.

D'autre part je regrette qu'il n'y ai pas plus de franchise entre nous et que certains soient incapables de me dire en face ce qu'ils pensent. N'oubliez pas que ce ne sont généralement pas ceux qui ont étéconcernés par les dessins qui ont "ralé" . J'aimerais avoir l'avis de Frédéric Poupon (JPC V1N3P25); Marco Benedetti (JPC V1N4P39); Jean Boschat (V1N6P3); Jacques Vaucelle (V1N7P30); et pour le prochain numéro,Jean Claude Becker. Doit-on aussi éliminer le peu d'humour qu'il y a: voir JPC V1N2P8, concernant un choc en retour à Jacob Daniel et écrit par Robert Schwartz BRAVO PEPE, n'oubliez pas que Jacob n'est pas trés vieux, il est étudiant et a passé son bac l'an dernier. Que ceux qui sont d'accord avec moi écrivent leurs articles d'une façon humoristique. Et que ceux qui ne désirent aucun dessins à leur sujet, m'écrivent; je ne transmettrai pas leur nom au journal et seront assurés de garder l'anonimat.

Dans l'espoir d'un meilleur échange, je vous souhaite à tous une Heureuse programmation.

PHILIPPE

P.S.: Je tiens à demander à Laurent Istria de m'excuser, car j'ai recopié son article (JPC V1N6P43) avec quelques fautes d'orthographe. Je tiens à signaler que je ne suis pas expert en machine à écrire et qu'il m'arrive très souvent d'en faire, car non seulement j'écris souvent mes articles le soir et il se peut que je sois un peu fatigué mais aussi je ne me relis jamais.

Merci de votre compréhension

```
01♦LBL "KAX"
"ANUM" CF 25 PASN
"×iu♦" RCL [ R↑ XTOA
R↑ XTOA RCL ' STO ]
X<> [ "⊢♦♦♦ B" X<
> ]
X<> [ STO \ "⊢"
X<> ] R↑ X<> c RCL \
, SIGN

25♦LBL 01
X<> IND L X<> [ "⊢*"
STO \ "⊢***" X<> \
"⊢***" X≠Y? X<> \
X=Y? SF 25 X=Y? R↑
"⊢****" STO ] "⊢♦♦"
X<> ] STO IND L
FC?C 25 ISG L X=Y?
GTO 01 R↑ STO c CLST
.END.
```

CODES DEC

P06: 245 1 105 12 0 240
P15: 247 127 0 0 0 240
166 66
P19: 242 127 240
P42: 244 127 0 0 240
P29, 31, 33: ACCESSIBLES DIRECTEMENT AU CLAVIER.

```
247 ENTER↑
82 ENTER↑
12 SF 27
XEQ "KAX"
01 W
LIST 001
01 "♦R♦♦♦♦"
```

```
01♦LBL "MKX"
CF 25 "%" PASN RCL '
R↑ CLA XTOA R↑ XTOA
RCL Z STO ] X<> [
"⊢♦♦♦♦aL" X<> ] X<> [
STO \ "⊢*" RCL ] SIGN
RCL \ RCL c X<> [
"⊢♦♦♦♦♦" X<> [ X<> d
CF 00 CF 01 CF 02
CF 03 X<> d "@×i×"
X<> [ STO \ ASTO X
X<> c 176 X<> Z SF 22

40♦LBL 00
RCL IND Z "**" X<> \
X<> [ STO ] ASTO [
ASHF X<> \ ASHF X≠Y?
X<> [ X=Y? CF 22 X=Y?
X<> L "⊢♦♦♦♦" STO \
"⊢♦♦♦" X<> [ STO ]
ARCL c X<> ] STO IND T
RDN FS? 22 ISG Z X=0?
GTO 00 X<>Y X<> c
CLST CLA CLD RTN
.END.
```

CODES DEC

P14: 247 127 0 0 0 0 4
76
P24: 246 127 0 0 0 0 0
P32: 244 64 1 105 1
P56: 245 127 0 0 0 0
P58: 245 127 0 0 0 240
P18, 42: ACCESSIBLES DIRECTEMENT AU CLAVIER.

=

```
247 ENTER↑
82 ENTER↑
12
XEQ "MKX"
W
75 W
LIST 001
75 "♦R♦♦♦♦"
```

LBL délocalisés en CB

L'utilité des LBL délocalisés n'est plus à démontrer et les possesseurs de lecteurs de CB vont pouvoir créer ces LBL ainsi que les GTO & XEQ correspondants ■■ moyen des CB ci-joints.

L'utilisation est identique à celle des ASN :
en MODe RUN (PRGM OFF)
XEQ ALPHA A ALPHA réponse: NONEXISTENT ("normal!")
MOD PRGM balayer les CB GTO^T ou XEQ^T et vous obtenez GTO "A" ou XEQ "A", ceci n'étant qu'un exemple.

POur les LBL, faire la même procédure indiquée ci-avant ■■ mode RUN puis ■■ MOD PRGM balayer les CB LBL^T, vous obtenez à l'AFF: pp ■ __
taper sur la touche LN (par exemple) et vous obtenez le LBL désiré.

Comme avec les Fonctions ASN, il est possible d'utiliser les CAR non accessibles directement au clavier :
placer ■■ ALPHA le CAR désiré (ou les CAR) :
CLA 114 XTOA 83 XTOA ASTO X
XEQ IND ■ réponse NONEXISTENT
MOD PRGM balayer les CB .....

Avotre disposition pour la création de tout CB.
Bonnes PRGMmations avec ou ■■■ LBL délocalisés,
rS (P20)

GTO^T

XEQ^T

LBL^T

W^T (?)

QL

BJ

Fonctions codées sur 2 Octets de CB

Page 141 du manuel PLOTTER, le codage des XROM des F est décrit à peu près de la façon suivante : (XROM 18,11)

```
                 18        11
BIN     ØØØØ 1 1ØØ1Ø ØØ1Ø11
HEX     Ø      C     ■     B
DEC          12         139
        ØØØØ 1 fffgg hhhhhh
```

où les 4 premiers BITs sont réservés pour la ■■■■ de contrôle (sigma des VAL DEC des 3 ½ O suivants, MOD 15 et = 15 si = Ø),

le 5ème BIT est 1 pour les XROM et ■ pour les autres F, les CAR ALPHA, les F indirectes et les F non PRGMmables (cf P217).

fffgg constitue le préfixe et nous pouvons donc en déduire que 31 est la VAL maximale codable et nous remarquons que 31 est également la VAL max du préfixe d'une XROM;

hhhhhh contient le suffixe dont la VAL max est donc 63 qui est également la VAL max du suffixe d'une XROM.

Nous pouvons en déduire que les VAL DEC de ØØØØ 1 fff sont des entiers positifs compris entre 8 et 15 et que gg hhhhhh est compris entre Ø ■ 255.

Les "matheux" en déduiront les 2 formules mathématiques permettant, avec 1 et Ø comme 5ème BIT, de trouver :
ØØØØ 1 fffgg hhhhhh ⟶ ØØØØ1fff gghhhhhh
ØØØØ Ø fffgg hhhhhh ⟶ ØØØØØfff gghhhhhh .

Pour les F indirectes, fff est 2 (Ø1Ø) et pour les F non PRGMmables, fff est 4 (1ØØ).

Pour les CAR ALPHA, fff est 1 (ØØ1) et gghhhhhh contient le CODE ASCII du CAR représenté d'une façon particulière :

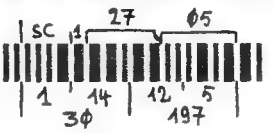
la VAL DEC du 1er ½ O est doublée; c'est tout du moins ce que j'ai cru comprendre avec mon "américain" minable (?).

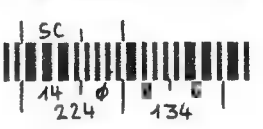
8-bit ASCII with Most Significant 4 Bits Doubled (For example the ASCII for "A" is 01000001 ; With most significant nybble doubled, the bar code value for the character A becomes 10000001.)

J'ai donc cru comprendre qu'il fallait doubler la VAL des 4 BITs de poids fort, cette dernière expression étant qqfois utilisée (?).

Je demande à ceux qui maîtrisent cette langue, de confirmer ou d'infirmer cette hypothèse.

J'espère que vous ne dormez pas et donc, merci de votre attention et à bientôt (?) rS (P20)





CB pour données ALPHA & APPEND ALPHA

Le 1er O contient la somme de contrôle de la même façon que pour les CB du type 6 (NUM).

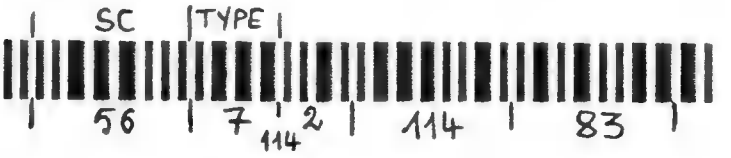
Le 3ème ½ O contient le type de CB : 7 pour ALPHA & 8 pour APPEND ALPHA.

Le 4ème ½ O contient le nbre de CAR de la chaîne.

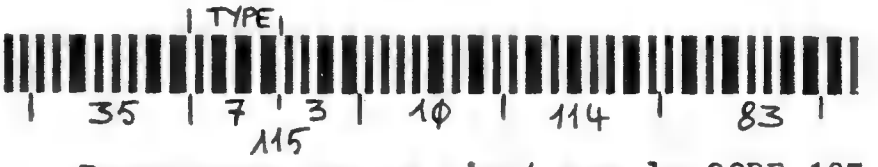
Les O suivants contiennent les CODES DEC des CAR (ASCII) et, compte tenu de la capacité de la MEM tampon du crayon, 14 CAR au maximum peuvent être inserrés.

L'examen des exemples ci-après sera plus explicite que je pourrais l'être.

Signalons simplement que seuls les CAR correspondant aux CODES Ø à 127 sont pris correctement en considération et que, avec l'IMP "IL", qq CODES ne nous donnent pas le résultat espéré pour la raison que je vous ai déjà indiquée.



r S



Remarquons que ce n'est pas le CODE 127 qui crée le APPEND mais le type de CB (8/3ème ½ O).

Les CODES inaccessibles directement au clavier peuvent être obtenus avec XTOA.

Aucun type de CB n'est prévu pour créer l'équivalent de la F WSTS mais rassurez-vous, avec un peu d'imagination (?), il est parfaitement possible de contourner le problème. Si cela vous intéresse, je vous indiquerais comment je procède mais vous pouvez chercher de votre côté; c'est assez passionnant.....

```
01♦LBL "PD"
02 XEQ "2D"
03 16
04 XEQ "QR"
05 LASTX
06 X↑2
07 *
08 RCL [
09 *
10 7
11 *
12 +
13 CLA
14 RTN
15♦LBL "QR"
16 X<>Y
17 STO ]
18 X<>Y
19 MOD
20 ST- ]
21 LASTX
22 ST/ ]
23 CLX
24 X<> ]
25 X<>Y
26 RTN
27♦LBL "2D"
28 "*"
29 X<> [
30 X<> /
31 ASHF
32 "⊢♦↓♦"♦♦
33 X<> [
34 X<> /
35 X<> [
36 "⊢♦♦B"
37 RCL [
38 INT
39 *
40 RCL /
41 *
42 ST+ [
43 X<> /
44 RCL ]
45 INT
46 HMS
47 *
48 RCL ]
49 +
50 E1
51 ST* [
52 *
53 X<> [
54 END

16 REG
109 OCTETS
CODES DEC :
P32: 247 127
     0 7 0
     96 0 8
P36: 244 127
     0 0 5
```

```
PRGM "COMP"
(JPC N5P28129)
LIGNES A MODIFIER POUR
COMPILER LES "LBL"

30♦LBL 10
40♦LBL 97
57 XEQ 93
73 XEQ 93
83 GTO 97
86 GTO 10
[illegible] GTO 97
303 GTO 01
306♦LBL 01
323 [illegible] 93
330♦LBL 93
333 GTO 93
338 GTO 11
342 GTO 02
343♦LBL 11
350♦LBL [illegible]
358♦LBL [illegible]
363 GTO 02
367 GTO 95
387 GTO 95
445♦LBL [illegible]
449 GTO [illegible]
468 XEQ 93
[illegible] XEQ 93
```

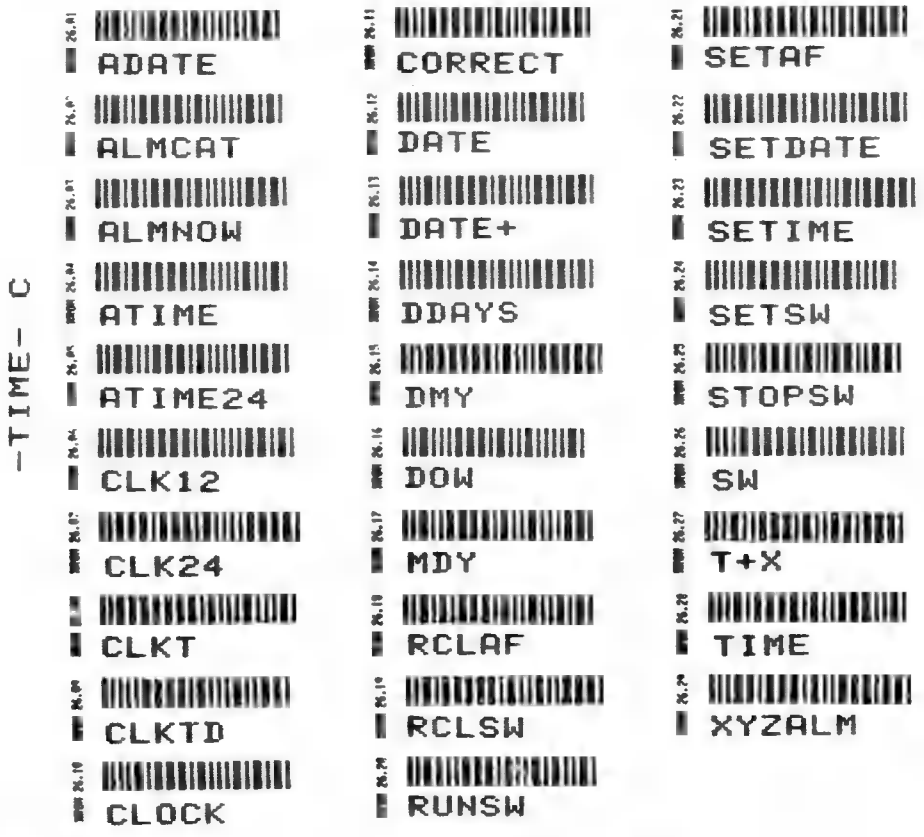
```
GTO .036
RCL b
XROM "PD"
2.600 [illegible]
GTO .088
RCL b
XROM "PD"
2.500 ***
2.600 X<>Y
94 ***
```

```
LBL "PD"
END
2 XEQ 06 01 PAS [illegible] LBL
210 BYTES
07 BYTES
CODE 1
[illegible] FREEMEMORY
```

```
RAP ..
01♦LBL "PP"
XEQ 06
LASTX
CLX - 2 / 3
STO 4 XEQ 06 X<>Y
FIX 4 XEQ 06
RCL X [illegible] 256 X<>Y
CLX RCL 1 X<>Y SF 00
IMI LASTX
FSC? 01 CLX
STO 0 X=0? SF 07
SF 06 X<>Y
FS? 14 SF [illegible] FS? 15
SF [illegible] FS? 13 SF 04
RCL 1 X<>Y
MOD 256 [illegible] 6
X<>Y STO 1 -6
X<>Y
X10M 64 9 FS? [illegible] 6
X10M RCL 1 X<>Y *
STO 1 FS? 15 SF [illegible] .
X<>Y FS? 16 SF [illegible]
SF 05 FS? 18 SF 06
FS? 19 [illegible] 07 FS? 20
SF 08 FS? 21 SF [illegible]
FS? 22 SF [illegible] FS? 23
SF 11 X<>Y 0 STO 1
FS? 11 X<>Y 0 STO 1
SF 07 X<>Y FC? 07 CLX
X<>Y 0 256 X<>Y 0
RCL 4 STO 96 CLA PCLPS STO 1
RCL \ X<>Y
```

Index des fonctions

Numéro XROM			
26.01	ADATE	Ajoute un nombre représentant une date au registre ALPHA.	Page 20
26.02	ALMCAT	Fournit un listage des alarmes; lorsqu'elle est arrêtée, réaffecte les touches du clavier à des fonctions non programmables du Catalogue des alarmes.	Page 43
26.03	ALMNOW	Active les alarmes de commande périmées.	Page 52
26.04	ATIME	Ajoute un nombre représentant une heure au registre ALPHA.	Page 19
26.05	ATIME24	Ajoute un nombre au format 24 heures au registre ALPHA.	Page 20
26.06	CLK12	Fait passer l'affichage du calculateur au format 12 heures.	Page 14
26.07	CLK24	Fait passer l'affichage du calculateur au format 24 heures.	Page 14
26.08	CLKT	Fait passer le calculateur en mode affichage de l'heure seulement.	Page 16
26.09	CLKTD	Fait passer le calculateur en mode affichage de l'heure et de la date.	Page 16
26.10	CLOCK	Affiche l'horloge.	Page 16
26.11	CORRECT	Met l'horloge à l'heure et règle le facteur d'exactitude.	Page 58
26.12	DATE	Place le nombre représentant la date actuelle dans le registre X.	Page [illegible]
26.13	DATE+	Calcule une nouvelle date à partir d'une date connue et du nombre de jours.	Page 23
26.14	DDAYS	Calcule le nombre de jours entre deux dates.	Page 24
26.15	DMY	Commute le format de la date sur jour-mois-année.	Page 13
26.18	RCLAF	Rappelle le facteur d'exactitude de l'horloge.	Page [illegible]
26.19	RCLSW	Place l'heure du chronomètre dans le registre X.	Page 37
26.20	RUNSW	Déclenche le chronomètre.	Page 37
26.21	SETAF	Spécifie le facteur d'exactitude de l'horloge.	Page [illegible]
26.22	SETDATE	Met l'horloge à une date donnée.	Page 14
26.23	SETIME	Met l'horloge à une heure donnée.	Page 14
26.24	SETSW	Met le chronomètre sur l'heure de départ spécifiée.	Page 36
26.25	STOPSW	Arrête le fonctionnement du chronomètre.	Page 37
26.26	SW	Met le calculateur en mode chronomètre et réaffecte les touches du clavier aux fonctions non programmables du chronomètre.	Page 29
26.27	T+X	Règle l'heure de l'horloge à un facteur spécifié.	Page 17
26.28	TIME	Place le nombre représentant l'heure actuelle dans le registre X.	Page 18
26.29	XYZALM	Enclenche l'alarme.	Page 41
	ON	Affiche l'horloge. Non programmable	Page 16
	DOW	26.16	
	MDY	26.17	



XROM 4,0 XROM 4,1
XROM 4,2 XROM 4,3
XROM 4,4 XROM 4,5
XROM 4,6 XROM 4,7
XROM 4,8 XROM 4,9
XROM 4,10 XROM 4,11
XROM 4,12 XROM 4,13
XROM 4,14 XROM 4,15
XROM 4,16 XROM 4,17
XROM 4,18 XROM 4,19
XROM 4,20 XROM 4,21
XROM 4,22 XROM 4,23
XROM 4,24 XROM 4,25
XROM 4,26 XROM 4,27
XROM 4,28 XROM 4,29
XROM 4,30 XROM 4,31
XROM 4,32 XROM 4,33
XROM 4,34 XROM 4,35
XROM 4,36 XROM 4,37
XROM 4,38 XROM 4,39
XROM 4,40 XROM 4,41
XROM 4,42 XROM 4,43
XROM 4,44 XROM 4,45
XROM 4,46 XROM 4,47
XROM 4,48 XROM 4,49
XROM 4,50 XROM 4,51
XROM 4,52 XROM 4,53
XROM 4,54 XROM 4,55
XROM 4,56 XROM 4,57
XROM 4,58 XROM 4,59
XROM 4,60 XROM 4,61
XROM 4,62 XROM 4,63

XROM 5,0
XROM 5,1
XROM 5,2
XROM 5,3
XROM 5,4
XROM 5,5
XROM 5,6
XROM 5,7
XROM 5,8
XROM 5,9
XROM 5,10
XROM 5,11
XROM 5,12
XROM 5,13
XROM 5,14
XROM 5,15
XROM 5,16
XROM 5,17
XROM 5,18
XROM 5,19
XROM 5,20
XROM 5,21
XROM 5,22
XROM 5,23
XROM 5,24
XROM 5,25
XROM 5,26
XROM 5,27
XROM 5,28
XROM 5,29
XROM 5,30
XROM 5,31
XROM 5,32
XROM 5,33
XROM 5,34
XROM 5,35
XROM 5,36
XROM 5,37
XROM 5,38
XROM 5,39
XROM 5,40
XROM 5,41
XROM 5,42
XROM 5,43
XROM 5,44
XROM 5,45
XROM 5,46
XROM 5,47
XROM 5,48
XROM 5,49
XROM 5,50
XROM 5,51
XROM 5,52
XROM 5,53
XROM 5,54
XROM 5,55
XROM 5,56
XROM 5,57
XROM 5,58
XROM 5,59
XROM 5,60
XROM 5,61
XROM 5,62
XROM 5,63

XROM 6,0 XROM 6,1
XROM 6,2 XROM 6,3
XROM 6,4 XROM 6,5
XROM 6,6 XROM 6,7
XROM 6,8 XROM 6,9
XROM 6,10 XROM 6,11
XROM 6,12 XROM 6,13
XROM 6,14 XROM 6,15
XROM 6,16 XROM 6,17
XROM 6,18 XROM 6,19
XROM 6,20 XROM 6,21
XROM 6,22 XROM 6,23
XROM 6,24 XROM 6,25
XROM 6,26 XROM 6,27
XROM 6,28 XROM 6,29
XROM 6,30 XROM 6,31
XROM 6,32 XROM 6,33
XROM 6,34 XROM 6,35
XROM 6,36 XROM 6,37
XROM 6,38 XROM 6,39
XROM 6,40 XROM 6,41
XROM 6,42 XROM 6,43
XROM 6,44 XROM 6,45
XROM 6,46 XROM 6,47
XROM 6,48 XROM 6,49
XROM 6,50 XROM 6,51
XROM 6,52 XROM 6,53
XROM 6,54 XROM 6,55
XROM 6,56 XROM 6,57
XROM 6,58 XROM 6,59
XROM 6,60 XROM 6,61
XROM 6,62 XROM 6,63

XROM 7,0 XROM 7,1

XROM 7,2 XROM 7,3

XROM 7,4 XROM 7,5

XROM 7,6 XROM 7,7

XROM 7,8 XROM 7,9

XROM 7,10 XROM 7,11

XROM 7,12 XROM 7,13

XROM 7,14 XROM 7,15

XROM 7,16 XROM 7,17

XROM 7,18 XROM 7,19

XROM 7,20 XROM 7,21

XROM 7,22 XROM 7,23

XROM 7,24 XROM 7,25

XROM 7,26 XROM 7,27

XROM 7,28 XROM 7,29

XROM 7,30 XROM 7,31

XROM 7,32 XROM 7,33

XROM 7,34 XROM 7,35

XROM 7,36 XROM 7,37

XROM 7,38 XROM 7,39

XROM 7,40 XROM 7,41

XROM 7,42 XROM 7,43

XROM 7,44 XROM 7,45

XROM 7,46 XROM 7,47

XROM 7,48 XROM 7,49

XROM 7,50 XROM 7,51

XROM 7,52 XROM 7,53

XROM 7,54 XROM 7,55

XROM 7,56 XROM 7,57

XROM 7,58 XROM 7,59

XROM 7,60 XROM 7,61

XROM 7,62 XROM 7,63